RISDM: A Requirements Inspection Systems Design Methodology

Perspective-Based Design of the Pragmatic Quality Model and Question Set to SRS

Shinobu Saito, Mutsuki Takeuchi
R&D Headquarters
NTT DATA Corporation
Tokyo, Japan
{saitousnb, takeuchimtk}@nttdata.co.jp

Setsuo Yamada
Software Innovation Center
NTT Corporation
Tokyo, Japan
yamada.setsuo@lab.ntt.co.jp

Mikio Aoyama
Dep. of Software Engineering
Nanzan University
Seto, Japan
mikio.aoyama@nifty.com

***Abstract*— The quality of the SRS (Software Requirements Specification) is the key to the success of software development. The inspection for the verification and validation of SRS are widely practiced, however, the techniques of inspection are rather ad hoc, and largely depend on the knowledge and skill of the people. This article proposes RISDM (Requirements Inspection Systems Design Methodology) to design the RIS (Requirements Inspection System) to be conducted by a third-party inspection team. The RISDM includes a meta-model and design process of RIS, PQM (Pragmatic Quality Model) of SRS, and a technique to generate inspection question set based on the PQM and PBR (Perspective-Based Reading). We have been applying the RIS designed by the proposed RISDM to more than 140 projects of a wide variety of software systems in NTT DATA for five years. By analyzing the statistics from the experience, we discovered some key quality characteristics of SRS reveal strong correlation to the project cost and level of quality to be used for evaluating the maturity of the SRS and predicting the risk.**

***Index Terms*—Requirements Inspection, Requirements Verification and Validation, SRS, Pragmatic Quality Model, Question Set, Risk Prediction.**

I. INTRODUCTION

To assure the quality of the SRS, the inspection and review[1] are the most effective and yet common practice in the V&V (Verification and Validation) process [2, 4, 21]. However, a recent survey revealed the practices of the requirements inspection are still rather ad hoc, although many projects and organizations defined the rules and procedures for the requirements inspection [12]. The survey also pointed out that lack of the methodology for systematic design of inspection method is one of the root causes of the poor practice of the requirements inspection. Therefore, the performance of the requirements inspection largely depends on the knowledge and skill of the people.

Many projects develop their own techniques apart from the organizational standard procedure. These poor practices lead to poor quality assurance of the requirements specifications, and bottles up the improvement of inspection techniques.

It is commonly practiced to divide of work in software development to maximize the ability of professionals. Therefore, in many projects we involved, the requirements specifications are developed by separate teams or organizations from the development teams and organizations. In such cases, it is critical to assure the quality of the SRS so that the development teams/organizations can understand the correct requirements and implement them as a software system.

The authors developed a third party inspection method and applied to real projects for several years [26]. However, the inspection method is specific to our organization and development context, and hard to apply to other organizations or different context. Therefore, it is necessary to develop a design methodology of requirements inspection method to be widely used.

This article proposes RISDM (Requirements Inspection System Design Methodology), pronounced like rhythm, a methodology to design RIS (Requirements Inspection System) to be conducted by a third party inspection team. Here, we call RIS as a system of conducting inspections to SRS with well-defined procedure and a set of coordinated techniques. Therefore, the proposed design methodology can create a RIS to SRSes for a project.

The major contribution of the proposed RISDM is to enable a systematic design of a RIS by integrating the following techniques;

- A meta-model of the RIS and generic process for designing a RIS based on the meta-model,
- PQM (Pragmatic Quality Model): A quality model of requirements specification from a specified viewpoint of a requirements reader,
- A technique to generate the question set for inspecting an SRS based on PBR (Perspective-Based Reading),
- A set of key quality factors, identified from the five years practice of the proposed methodology, which are useful to improve the SRS and predict the risk in terms of the cost-overrun of the project.

[1] We use inspection due to the formality and rigorousness [16].

The rest of this article is structured as follows.

Chapter II discusses related works. Based on the research context and approach illustrated in Chapter III, the RISDM (RIS Design Methodology) is proposed in Chapter IV. Chapter V reports the current status of the practice of the RIS designed by the proposed methodology. Chapter VI discusses the experiences of the practice of the RIS, and reveals some key quality characteristics to improve the quality of the SRS and evaluate the risks in terms of cost overrun of the project. Chapter VII discusses the contributions of this work, followed by the conclusions and future works in Chapter VIII.

II. Related Work

A. *Standard and Quality Model of SRS*

IEEE 830 is a widely recognized standard of SRS [15]. It also defines eight quality characteristics of an SRS. However, the quality characteristics are rather generic by its definition, and hard to apply in practical use. Furthermore, the contents of an SRS are diverse; ranging from business requirements to software requirements. In practice, it is necessary to provide a concrete definition of quality characteristics of an SRS by considering its scope and context.

Davis et al. proposed 24 attributes of software requirements specifications [6]. For more concrete quality model, Krogstie et al. defined "pragmatic quality" as a quality specific to a certain reader, say, user or developer to the SRS [20]. Fabbrini et al. extended the pragmatic quality model, and defined four layers of quality types of syntactic, structural, semantic, and pragmatic for the SRS written by natural language [9]. The pragmatic quality model is promising to evaluate the quality of SRS due to the diversity of the readers. However, conventional works do not address the techniques for the design and use of the pragmatic quality model [25]. It is necessary to develop a practical method to evaluate the pragmatic quality of the SRS.

B. *Inspection Techniques of SRS*

Inspection is well-defined discipline, and has been widely applied in the implementation process of software development [10]. In requirements engineering, there are a bunch of literatures on the inspection and review [1, 2, 4, 13, 16, 17, 18, 24, 27, 28]. For example, Pohl introduces four classifications of techniques of validating SRS [23]. However, the inspection and review of SRS need special attention [18]. Syntactically, it has to deal with diverse descriptions in natural language text, figures and tables. Semantically, the contents of the SRS are diverse, and need rich knowledge of the domain and context of the objective system ranging from business to software. Sommerville and Sawyer suggest to organize formal requirements inspection [30]. However, inspection of SRS in practice is time consuming and error prone, largely depends on human knowledge and reading skill [13].

On the other hand, techniques of SRS inspection are less mature than source code inspection due to the diversity of SRS [28]. To cope with the diversity, PBR (Perspective-Based Reading) is a promising technique for SRS inspection [29]. It generates question set to a document from a specific perspective, say, user, or developer. Inspectors can read the document with the question set. However, how to design the question set is left to practitioners. The authors also developed a third-party inspection technique based on the PBR and pragmatic quality model, and practiced it to a number of real projects [26]. However, the proposed technique weak in the systematic design of question set, and requirements inspection system.

C. *Impact of SRS Inspection and Quality*

It is widely acknowledged that the quality of SRS is the key to success or failure of the subsequent development [7, 19, 21]. However, the evidence of the impact of quality of the SRS to the subsequent development is still scarce.

One of the early work is the analysis of statistical of NASA. It revealed a negative correlation between percent project cost for RE and percent project cost overrun [11]. Damian et al. empirically studied the complex impact of RE to other processes with the statistics collected from the commercial software development [5]. Recently, the authors revealed ROI of RE, that is, the ratio of reduction of total development cost over the requirements inspection cost [19, 25]. As a related work, measuring SRS in text is also important, but is a difficult issue [13]. Another approach is the five levels of taxonomy of the impact of RE process change [14].

All the related works touch some aspects of the quality and impact of the SRS to the subsequent development. However, we need more comprehensive analysis of the quality of the SRS and its impact to the entire development. Furthermore, practitioners require useful measure such as how the quality of SRS impacts to the cost, quality and delivery of the system, which is largely unknown in the literatures.

III. RESEARCH CONTEXT AND APPROACH

A. *Research Context:A Third-Party Inspection Process*

Figure 1 illustrates the key ideas of the proposed design

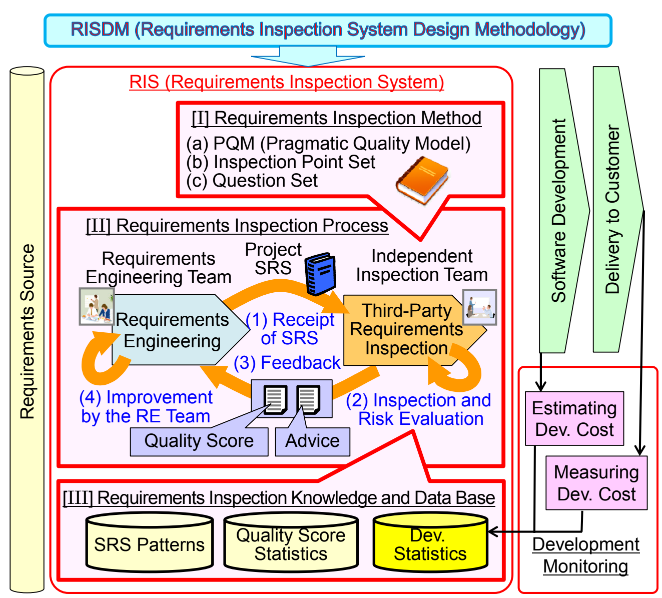

Fig. 1. RISDM and RIS for Third-Party Inspection

methodology for the RIS, which is assumed to apply to the third-party inspection process we proposed [26]. We assume two key stakeholders of RE (Requirements Engineering) teams and a third-party inspection team. The inspection team is responsible to inspect all the SRSes. The third-party inspection process consists of the following four processes.

1) Receipt of SRS: The inspection team receives an SRS from the requirements engineering team.

2) Inspection and Risk Evaluation: The inspection team inspects the SRS with the RIS, which should be designed in advance with the design methodology proposed. The inspection team compiles the inspection report, including QS (Quality Score) report and improvement advice. The QS report includes benchmarking of the QS of the SRS with QS statistics of other SRSes, and SRS patterns. Based on the QS, the team also evaluates the risk of the subsequent development. A set of advices is compiled to improve the quality of the SRS, and requirements engineering process of the team as well.

3) Reporting with Evaluation and Feedback: Based on the inspection report, the inspection team has a meeting with the RE team, and discuss possible improvements.

4) Improvement done by the RE Team: The RE team takes necessary actions based on the inspection report. If the quality of the SRS is concerned, the RE team revises the SRS and re-submit it to the inspection team for re-inspection.

The objective of this research is design methodology of RIS, a system of inspection. We use system for inspection since we view inspection as an organized and systematic activities to ensure the quality of the SRS. Therefore, we need to design the process and data model for the RIS just like design software systems.

B. Key Ideas for the RIS (Requirements Inspection System)

The basic idea of this work lies in that the requirements inspection is a system as a whole much like software development. Therefore, we can design the RIS (Requirements Inspection System), like we design a software engineering process and methodology with a reflection of "software process as a software". As a design methodology of RIS, we propose

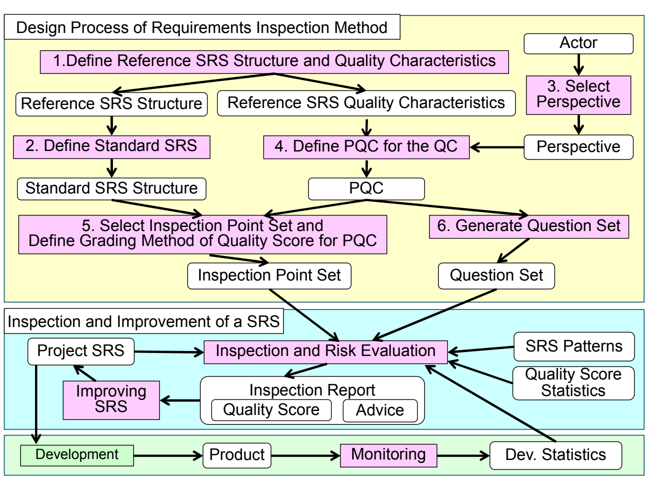

Fig. 2. Design Process of Requirements Inspection Method

RISDM (RIS Design Methodology), pronounced like rhythm, in this article.

As illustrated in Figure 1, the RIS consists of three key technologies indicated by three boxes:

- [I] Requirements Inspection Method,
- [II] Requirements Inspection Process, and
- [III] Requirements Inspection Knowledge and Data Base.

Requirements Inspection Method is a set of well-organized concrete techniques for the inspection. For the Requirements Inspection Method, we developed three key techniques including

- (a) PQM (Pragmatic Quality Model),
- (b) Inspection Point Set, and,
- (c) Question Set.

PQM is a comprehensive organization around PQC (Perspective Quality Characteristics), which is derived from Perspective and Reference QC (Quality Characteristics).

The core idea of Requirements Inspection Method comes from two ideas of PBR (Perspective-Based Reading) and Pragmatic Quality.

PBR helps to narrow down reading perspective to a specific perspective of the intended readers of the SRS. The Pragmatic Quality is an idea of quality for certain user of a product in a certain context. In the RIS, the pragmatic quality, we mean, a quality of the SRS for a user in the context of reading SRS. The PQC is a specialization of general QC, such as those defined in IEEE Std. 830. In this article, we propose a technique to derive PQC from the QC for certain Perspective, since the Perspective for PBR has to consistent with the Perspective to PQR. The Perspectives include customer/user and developer of the system to be developed.

Inspection Point Set is a set of points on the specific element of SRS to be inspected from a specific Perspective.

Question Set is a set of questions at the Inspection Point regarding on the each element of PQC.

By integrating the three techniques of PQM, Inspection Point Set, and Question Set. The Requirements Inspection Method enables the inspection team of little knowledge on domains to inspect diverse SRSes with productivity and quality of inspection.

Requirements Inspection Knowledge and Data Base provide information to the inspection team for inspection and evaluation. The Cost Statistics DB stores performance statistics of each project. It includes performance of both requirements engineering and subsequent development processes. As discussed later, we can take advantage of using the statistics for not only evaluate the quality of the SRS but also evaluate the risk of the subsequent development based on the quality of the SRS and statistics from the DB.

The results of the evaluation are compiled into the inspection report to the RE team. The inspection report also includes benchmarking of the quality of the SRS against the statistics of all the previous projects, and improvement advice with useful SRS patterns.

IV. Design Methodology for RIS

A. Design Process of RIS

Figure 2 illustrates the design process of the RIS and execution of an instance of the RIS designed for inspection and improvement of Project SRSes in a specific project. The instance of the RIS is a part of the entire software development process as indicated in the bottom part of the figure.

By its definition, the design process, we propose, is expected to be executed before enacting the RIS. The design process consists of seven activities, and generates the PQM, the inspection point set, and the question set. An inspection team can systematically inspect an SRS by using the three techniques. The inspection team evaluates the quality of the SRS with QS (Quality Score) and compiles advices based on the QS. The QS and advices are compiled into an inspection report, and fed back to the RE team.

The inspection team refers the SRS patterns extracted from the best practice of SRSes, and statistics of QS. The RE team can improve the SRS based on the inspection report.

One of the uniqueness of the RIS system illustrated in Figure 1 is its capability to evaluate the risk of the subsequent development in terms of cost and schedule overrun based on the quality of SRS and development statistics collected.

B. Information Model of Requirements Inspection Method

Requirements inspection method is the heart of the RIS. Figure 3 illustrates an information model of the requirements inspection method corresponding to the design process illustrated in Figure 2.

There are three roots entities in the information model; Reference SRS Structure, Reference SRS QC (Quality Characteristics) and Perspective. The relationship of three root entities indicates the model of the requirements inspection method.

The Reference SRS Structure defines a TOC of SRS as a reference for the Project SRS to be inspected. The Reference QC defines generic quality characteristics for the SRS. The Perspective defined a specific view to the SRS for inspection. It might be a view from an actor, say a user or a designer. PQC (Pragmatic Quality Characteristics) is derived from the Reference QC for a specific Perspective.

The question set is a set of questions to inspect to an SRS from a perspective.

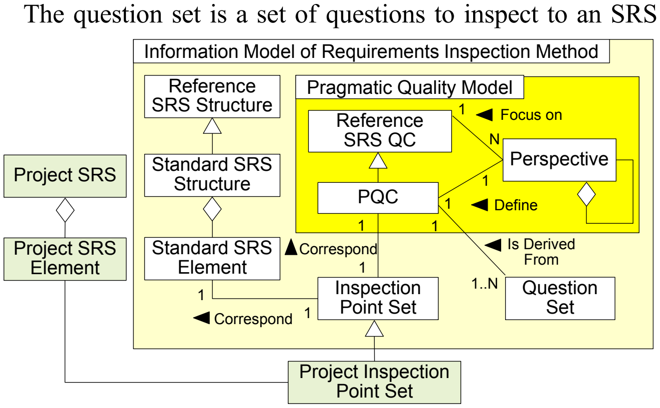

Fig. 3. Information Model of Requirements Inspection Method

A standard SRS is derived from Reference SRS for a specific organization. Inspection point defines whether an inspection is necessary at a cross section between each element of the standard and each characteristic of PQC.

The Inspection Point Set is a collection of inspection points over all the elements and PQC. It is specialized to the specific SRS of the specific organization.

C. Design Process

To explain the details of the RISDM process, we employ real cases at NTT DATA and walk through the process.

1) Define Reference SRS Structure and Quality Characteristics: We select the Reference SRS Structure and Quality Characteristics as a baseline of the requirements inspection method. Actually, we employed IEEE Std. 830 as for both Quality Characteristics and SRS Reference Structure. It is possible to select any Quality Characteristics and SRS Reference Structure other than IEEE Std. 830.

2) Define Standard SRS: Based on the SRS Reference Structure selected in Activity (1), we define the Standard SRS Structure specific to the organization, say NTT DATA. If the organization already defined a Standard SRS Structure, it is only necessary to map the Standard SRS Structure and the Reference SRS Structure. The mapping ensures the completeness of the Standard SRS Structure.

Table 1 exemplifies the Standard SRS Structure of NTT DATA. The Standard SRS Structure includes TOC and a set of

TABLE I. Table of Contents of Standard SRS

Chapter	Section
1. Introduction	1.1. Purpose of SRS
	1.2. Intended Reader
	1.3. Structure of SRS
	1.4. References
2. Overview of System	2.1. Goal of System
	2.2. Business and Scope of System
	2.3. Constraints
	2.4. Terms
3. Items Causing Change or Unspecified	3.1. Items Causing Changes
	3.2. Items Unspecified
4. Functional Requirements	4.1. Business Flow
	4.2. Functions
	4.3. Data Model Definition
5. Non Functional Requirements	5.1. NFR Grade
	5.2. Requirements to System Architecture
	5.3. Requirements to Migration
	5.4. Requirements to Service Provisioning

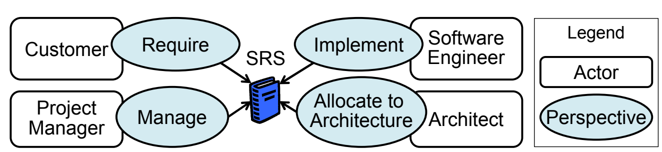

Fig. 4. Actors and Perspectives

TABLE II. PERSPECTIVES AND PQC

Perspective		Reference Quality Characteristics							PQC (Pragmatic Quality Characteristics)		
Level 1	Level 2	Correct	Complete	Ranked for importance and/or stability	Unambiguous	Verifiable	Traceable	Modifiable	ID	Name	Definition
Require	Conform to customer's needs	X							C1	Correspond-ence to goals	All business & system requirements should correspond to more than one project goal.
Manage	Manage all elements on the progress		X	X					C2	Coverability	All contents of SRS should be described.
Implement	Design by template				X	X			C3	Template usage	Template of SRS should be used.
	Design by standard description				X	X			C4	Standard description usage	Standard description (notation) of SRS should be used.
	Use with common terms				X				C5	Definition of terms	Terms of the project SRS and their definitions are should be created.
Allocate to Architecture	Overview all the elements							X	C6	Listing with identifier	All artifacts should be stated in the artifact lists and have identifier.
	Specify all the elements						X	X	C7	Identifiability	All artifacts and certain elements of the artifact are identifiable using identifier.

associated SRS templates. The SRS templates include a template of functional descriptions and UI template. Table 1 shows only two levels of chapter and sections. However, The Standard SRS Structure specifies further details.

3) Select Perspective: Define a perspective for review in order to define the PQC concretely. First, identify an actor for whom the inspection team inspects the SRS. Then, define a set of perspectives to the Standard SRS from the actor(s) identified. If necessary, the perspective identified can be decomposed into detailed perspectives.

Figure 4 exemplifies four actors and associated perspectives we identified. Since the perspectives are top level, we call them Level 1perspectives. The "Implement" and "Allocate to Architecture" perspectives are decomposed into detailed perspectives exemplified in the left column in Table 2.

4) Define PQC for the QC: Define PQC from the Reference QC (Quality Characteristics) in accordance with the Perspectives identified in Activity (3).

Table 2 exemplified the PQC. Here, we have four Perspectives at the top level. "Require" indicates a perspective from customer who require the requirements. It is refined to "Conform to customer's needs" at the second level of Perspective, Level 2. From the perspective, the "Correctness" QC needs to be assured, as indicated by "X" at the cross point of "Conform to customer's needs" perspective and "Correctness" QC. From the "Conform to customer's needs", the "Correctness" QC is refined to "C1: Correspondence to goals" of PQC. Like this, all the QC elements are refined to PQC elements.

Note that, by the definition, the granularity of PQC is same to that of Perspectives, namely, seven Level 2 perspectives.

We call the structure of Perspective, Reference QC and PQC as PQM (Pragmatic Quality Model) as illustrated in Figure 3.

5) Select Inspection Point Set and Define Grading Method of Quality Score for PQC: This activity is to select inspection points set and define grading method of quality score for evaluating the SRS based on the PQC allocated to the SRS.

Inspection point is an element of SRS where an inspection is needed with respect a certain element of the PQC. The idea of inspection point comes from the fact that not every element of PQC is meaningful in the inspection to every element of the Standard SRS. Therefore, it is necessary to clarify which PQC element is effective in inspection to which element of the Standard SRS. Therefore, we select meaningful elements of PQC to each element of the Standard SRS.

The results of this selection are represented by "Inspection Point Set" illustrated in Table 3. In the table, each row indicates an element of PQC, and each column indicates an element of Standard SRS. The marking "X" at the cross point

TABLE III. INSPECTION POINT SET

PQC		TOC of Standard SRS			
ID	Name	2.1 Goal of System	2.2 Business & Scope of System	2.3 Constraint	2.4 Term
C1	Correspondence to goals	X			
C2	Coverability	X	X	X	X
C1	Template usage	X	X	X	
C4	Standard description		X		
C5	Preparation of glossary				X
C6	Listing with identifier	X	X	X	
C7	Identifiability	X	X	X	X

is the inspection point which indicates inspection is effective and necessary. For example, “C1: Correspondence to goals”, an element of PQC, is effective and necessary to “Section 2.1 Goal of System”.

We identified 196 effective inspection points in our RIS.

Next, we need to define grading method of QS (Quality Sore) for the inspection point set. QS is a score representing the quality of the SRS. Since we design every question, allocated to each inspection point, is simple and can be answered by “yes” or “no”, each inspection point can be scored to 1 or 0. Therefore, the quality of the SRS can be graded by summing the score over the inspection point set including 196 inspection points in our case.

The grading method is up to the organization that uses the RIS. However, we define the following basic grading method

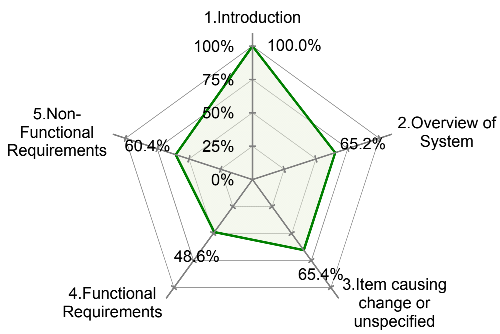

Fig. 5. An Example of Quality Score by Chapter

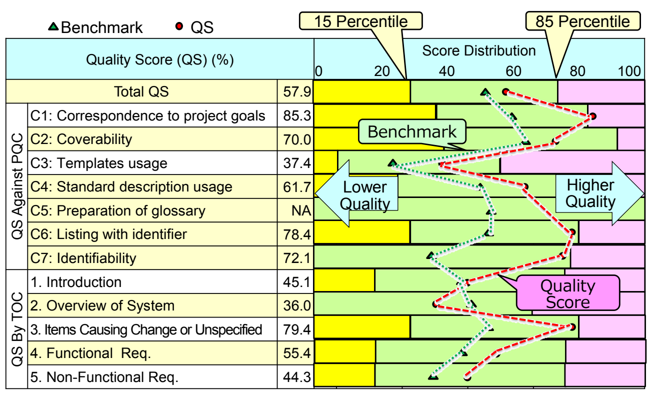

Fig. 6. Benchmarking with All Other Projects

based on the PQM.

a) Grading by PQC or Standard SRS Element: Sum of the score of the inspection by PQC or chapter of the SRS reveals the quality by PQC or chapter. We employ two types of grading.

i) TQS (Total Quality Score): By summing up the score and adjusting by the total number of the inspection points, we can evaluate an SRS by a simple scalar value of TQS ranging from 0 to 100.

ii) Distribution of PQC: As illustrated in Figure 5, a distribution of PQC shows the quality by chapter after adjustment with the baseline. A distribution by PQC illustrated in the upper half of Figure 6 indicates the strength and weakness of the SRS in terms of PQC.

b) Benchmarking: Figure 6 illustrates a benchmarking of an SRS against other project. The dotted line indicates the average score of all other projects and solid line indicated the score of the SRS. Note that the score is presented from two different views of PQC, C1 to C7, and chapter from 1to 5.

6) Generate Question Set: Define “Question Set” from the PQC as exemplified in Table 4. Note that an element of “Question Set”, that is “a question”, corresponds to an element of PQC.

The question set also serves a precise reading guide, with which the inspection team reads the SRS from the specified perspective. Therefore, we think the question should be fine grained according to the specific PQC element, and make the reader clearly judge whether the specific SRS element meet the quality criterion in terms of the PQC element.

We employed a style of a question for checking an SRS element against a PQC element since it is simple enough to judge the quality of an element of SRS clearly while providing enough information on the SRS element.

V. PRACTICE

A. Corporate-Wide Application

The RIS designed by the proposed RISDM is practiced by a third-party inspection team in NTT DATA since 2010 after intensive empirical evaluation of the RIS. NTT DATA is a global software and service provider with some 10,000 professionals in Japan, and more than 60,000 worldwide [22].

TABLE IV. QUESTION SET

PQC		Question Set	No. of points
ID	Name		
C1	Correspondence to goals	Are business and system requirements corresponding to the project goals?	3
C2	Coverability	Are there elements of the project SRS corresponding to the elements of the standard SRS?	54
C3	Templates usage	Are artifacts described using the template which is selected in the standard SRS?	36
C4	Standard description usage	Are artifacts described by the standard description which is selected in the standard SRS?	6
C5	Definition of terms	Are there glossaries of the project SRS created?	3
C6	Listing with identifier	Are artifacts and certain elements of the artifact given identifier and listed in the table?	48
C7	Identifiability	Are artifact and certain elements of the artifact identified using identifier?	46
		Total	196

As illustrated in Figure 7, the RIS is widely used by more than 140 projects in NTT DATA Japan. Now, it is one of the corporate wide standard processes for software development. The size of the projects inspected ranges from medium to large. Some 25% of the projects exceeds 500 person month of the efforts.

B. *Practice of Third-Party Inspection Team*

Within NTT DATA Japan, a third-party inspection team conducts the inspections based on the RIS. The team consists of three to four dedicated professionals and a few part time professional who can help the inspection team if the demand is too high. We provide training program for inspectors. It takes about one week for them to accquire the inspection method by the trainige program.

An inspection usually takes three weeks: one week for reading the SRS, the second week for analysis and evaluation of the SRS and quality score, and the third week for compiling an inspection report. The inspection team also monitors the development, and collect statistics.

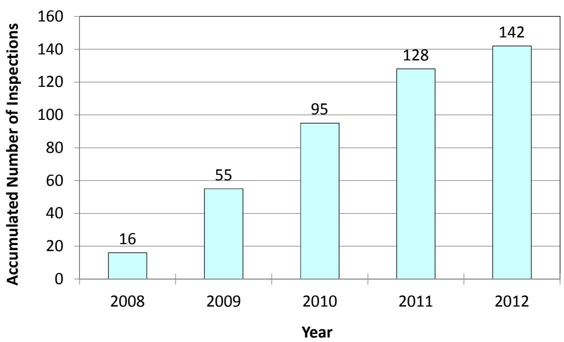

Fig. 7. Accumulated Numbers of Projects Inspected

TABLE V. PROJECT PROFILE FOR ANALYZING COST OVERRUN

ID	Domain	Development Type	SRS pages
1	Financial	Extension & Replacement	545
2	Public	New	401
3	Manufacturing	New	666
4	Public	New	172
5	Financial	Extension & Replacement	145
6	Distributor	New	300
7	Service	New	71

TABLE VI. CORRELATION COEFFICIENT AND P VALUE

PQC		Correlation Coefficient	P Value
C1	Correspondence to project goals	0516	0.236
C2	Coverability	-0.793	0.033
C3	Templates usage	-0.226	0.626
C4	Standard description usage	-0.219	0.638
C5	Preparation of glossary	-0.013	0.978
C6	Listing with identifier	-0.408	0,364
C7	Identifiability	-0.794	0.033

VI. EVALUATION

We found some useful characteristics of PQC in the practice of RIS designed by the proposed RISDM. To evaluate the RIS and associated PQM, we conducted an intensive analysis to statistics collected from the practice done by the third-party inspection team.

A. *PQM for Risk Evaluation*

1) Purpose: To demonstrate PQC is effective characteristics to predict risks of the subsequent development. The risk can be measured by cost, quality and time. We already identified impacts of the PQC to development time in terms of the delay of the delivery [25], and the impact of missing traceability. However, we do not have answers to the following questions.

- Is there any elements of PQC useful to predict any risks?, and
- Is there any elements of PQC which is more significant than others in predicting the risks such as cost overrun?

2) Method: As listed in Table 5, we collected statistics of seven projects, which are completed the entire development processes until 2013, and are complete in the sense that all the measures are completely measured throughout the development processes. The projects vary in domain, development type, and size in terms of number of pages of the SRS. The statistics include all the elements of PQC and cost of both estimated and actual the subsequent processes. The cost of subsequent development is estimated when the SRS is completed. The actual cost is fixed upon delivery. As illustrated in Figure 1, corporate wide project monitoring system collects the estimated and actual cost of each project and stores them to the development statistics DB. We referred the development statistics for the evaluation of CDR (Cost Difference Ratio), a ratio of estimated and actual cost, defined by the following equation (1).

$$\mathrm{CDR} = \frac{|\ \text{Actual Cost} - \text{Estimated Cost}\ |}{|\ \text{Estimated Cost}\ |} \qquad (1)$$

3) Results and Findings: Table 6 shows the results of correlation analysis between seven PQC of C1 to C7 and CDR. The table shows both correlation coefficient and P-value. It should be noted that the correlation coefficients of both C2 and C7 are less than – 0.5, which indicates strong negative correlation with CDR. If the quality score of C2 or C7 of an SRS is low, the project has a high risk of cost overrun. Especially, P-value of C2 is less than 0.1, and the correlation stands to the population at the level of 10% statistical significance.

Figure 8 illustrates a scatter diagram between the PQC of C2 and C7 and CDR for six projects listed in Table 7. Linear regression for C2 and C7 are very closed. The diagram is separated into two parts by 25% of CDR. We use 25% threshold of CDR from COCOMO II [3].

The linear regression for C2 and C7 cross the 25% threshold of CDR at 59.5% and 63.0%, respectively. Therefore, we assume 59.5% is the lowest threshold of PQC, and further

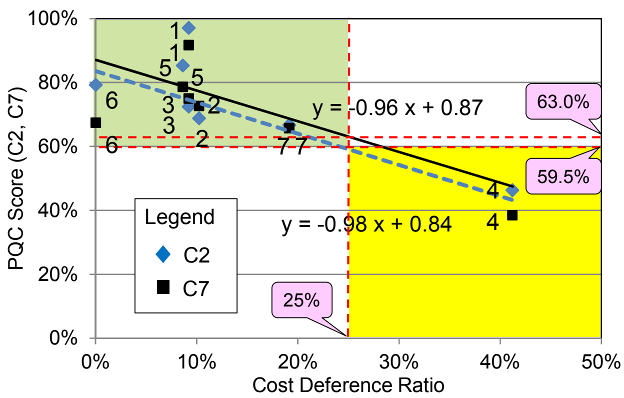

Fig. 8 PQC Score (C2, C7) and CDR

divide the diagram into four areas. Among them, two areas of upper left and lower right are meaningful. The upper left zone indicates higher quality of the SRS and lower risk of cost overrun of the project, while the lower right zone is opposite: the lower quality of the SRS and higher risk of the cost overrun of the project. This statistic is very useful to predict the risk of a project at the early stage of software development when the SRS is completed.

B. Effect of Improving the SRS by the Feedback

1) Purpose: To demonstrate how the feedback with the inspection report helps to improve the quality of an SRS.

2) Method: We monitored the change of PQC of the SRS from six projects listed in Table 7. The SRS of the projects were revised based on the inspection reports. We evaluated the PQC for the revised SRS, and analyzed the improvement process.

TABLE VII. PROJECT PROFILE FOR ANALYZING PQC IMPROVEMENT

ID	Domain	Development Type	SRS Pages
A	Financial	New	191
B	Manufacturing	New	289
C	Financial	Extension	550
D	Service	Replacement	343
E	Financial	Replacement	415
F	Financial	New	267

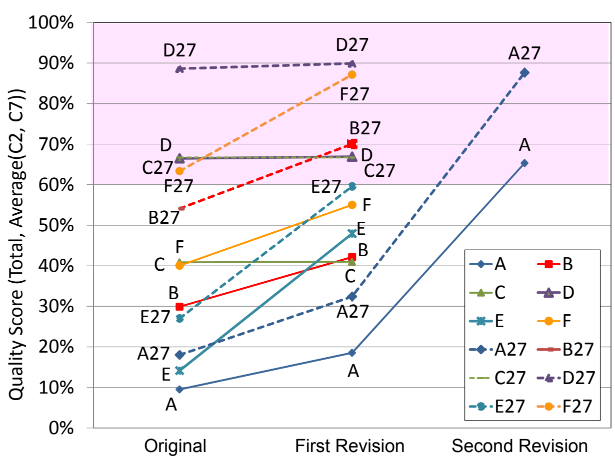

Fig. 9. Improvement of PQC

As Table 7 indicates, each project differs in the domain, development type, and size in terms of the number of SRS pages.

3) Results and Findings: Figure 9 illustrates the change of TQS (Total Quality Score) and average of C2 and C7 along with the revisions. Besides TQS, we employed the average of C2 and C7 since we found C2 and C7 are key quality characteristics by the statistical analysis discussed above.

Project A changed the SRS twice, while all other projects did once. The solid line indicates TQS while a dotted line does the average of C and C7, labelled by Project ID and "27".

From the figure, we found a "satisfaction criteria" of quality score.

We can observe the trend of improvement. The solid lines indicate four out of the six projects improve the quality, while two projects of C and D are almost unchanged. Among them,project C appears to be under 40% after the revision. However, looking at the dotted lines, the averages of C2 and C7 of projects C and D, C27 and D27, exceed 60%. The results can be interpreted that the two projects of C and D meet the certain level of quality, say more than 60%, with respect to C2 and C7 at the original SRS. Therefore, the PE teams of the projects did not think they need to further improve the SRS.

Other four projects, A, B, E, and F, did improve their SRS and attained the level of C2 and C7 over 60%. Like this, closer look to PQC of each project can reveal why and how each project improved, or did not improve, the quality of the SRS.

It is worth to note that after the improvement all the project lower the risk by attaining the average of C2 and C7 of SRS over the level of 60%.

We found that the improvement of PQC is not uniform as illustrated in Figure 10. The figure shows the distribution of PQC at three versions. The solid line indicates the distribution of the PQC of the original SRS reported in the inspection report. Based on the report, the project changes the SRS.

The dotted line indicates the PQC distribution of the first revision. Two quality scores of C2 and C4 are significantly improved while others are remained low. The improvement of the C2 means that the coverability is improved. Actually, the

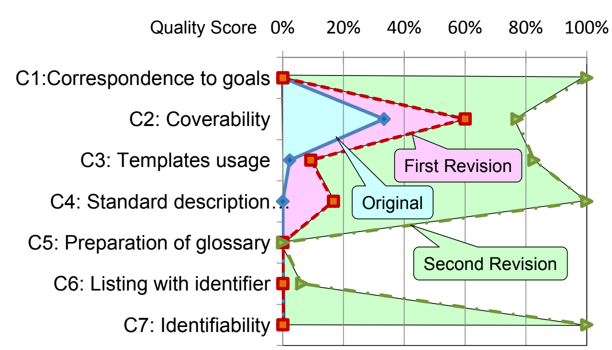

Fig. 10. Incremental Improvements of QS at Project A

number of pages of SRS was significantly increased from 106 to 191.

In the second revision, shown by dashed-dotted line, the C1, C2 and C7 are significantly improved. It means the quality of content, such as traceability, is semantically improved. Actually, the number of pages of the SRS was unchanged in this revision.

As a whole, both C2 and C7 are improved to attain the quality level of low risk.

From the detailed analysis, we found the process of how a project improved the quality of the SRS.

VII. DISCUSSIONS

A. *Feasibility of RISDM*

As discussed in the previous chapter, we believe he proposed RISDM is practically workable and useful. One of the contribution of the RISDM lies in that there is no design methodology for designing RIS, and many inspections, including RIS, are home grown and experience-based. Therefore, RIS is rather ad hoc, and differs project by project. Comparing with source code inspection, requirements inspection is much more difficult due to the diversity of its contents and writing styles. There is no consistent measure to measure the quality of SRSes across multiple projects.

Within the authors' knowledge, the proposed RISDM is first attempt to solve the problem. Although the practices are limited within a one company, the SRSes inspected exceeded 140, which are diverse in terms of size, domain, and writing styles. The success lies in the systematic design of the RIS with the proposed RISDM. Without the RISDM, we believe, it is still difficult to initiate a third-party inspection system for diverse applications.

It is worth to note that the RISDM is general enough to design different inspection system by changing the Perspective and PQC.

B. *Effectiveness of PQM*

Empirical study in the previous chapter discovered some nature of PQC (Pragmatic Quality Characteristics). Among them, we found two PQCs, Coverability (C2) and Traceability (C7) are influential to the quality of an SRS and subsequent software development process.

Poor coverability causes incompleteness of the SRS and makes the cost estimation inaccurate.

Poor traceability causes missing and/or misunderstanding.

If the evaluated score of C2 or C7 is low, it implies that risk of cost overrun significantly increased.

Therefore, by C2 and C7, we can predict the possible disaster, and help people to take any necessary strategy to avoid or mitigate the risks. This is particularly useful to the project manager and requirements analysts.

We also found that other PQC, such as "C4: Standard description usage" has less impact on the cost overrun of the project. However, such PQC may have an impact on other risks, which is not yet clarified.

C. *Effectiveness of the Feedback*

The in-depth analysis of the improvement of SRS, as illustrated in Figure 10, we found that the improvement of SRS is not uniform, and C2 and C7 are considered as key indicators to evaluate the improvement.

On the other hand, we found that 60% of the average f C2 and C7 is a lower threshold of the quality level as indicated in Figure 9. The fact is discovered by this study, and RE teams are not aware it. However, all the RE teams considered 60% as the lowest goal of the quality. This implies that improvement activities of the RE teams are reasonable and good enough.

It is worth to note that the question set is useful for an improvement guideline, although it is designed for evaluate the SRS.

D. *Integrated Methodology*

The RISDM is the design methodology for integrating pragmatic quality and PBR (Perspective Based Reading) for inspecting SRS. This methodology provided the systematic design process of requirements inspection method. It enables even third-party inspectors, who have less knowledge on project context and background, can conduct an inspection of project SRSs. With reference to the method defined by the RISDM, in NTT DATA, the third-party inspection process for SRSs of large scale software development projects could work in practice. As above mentioned, two approaches (i.e., pragmatic quality, PBR) have been studied independently. Each approach could make some contribution, but is limited to partial solution. Our technical contribution is that we have shown the integration of both approaches could elaborate a practical and effective methodology, namely RISDM. Moreover, the RISDM is unconstrained by existing domain knowledge. It could be applied to not only enterprise system development, but also embedded system, and package-based solution.

VIII. CONCLUSIONS AND FUTURE WORKS

This paper proposed RISDM (Requirements Inspection System Design Methodology), a methodology for designing the RIS (Requirements Inspection System). Our methodology provides a design process of requirements inspection method by integrating both approaches of pragmatic quality and PBR (Perspective Based Reading). In the process, we defined the PQM (Pragmatic Quality Model) which is composed of Perspective of certain reader of SRS, Reference QC (Quality Characteristic) of SRS, and PQC (Pragmatic Quality Characteristics) which are derived from Perspective and Reference QC. Next, the question set is generated from the set of PQCs. The question set makes the inspector clearly judge whether the specific elements of the project SRS meet the quality criterion in terms of the PQC element. We have designed and practiced the inspection method to an industrial inspection process. Moreover, after conducting the inspection of the project SRS, we have monitored the life cycle of the project and collected data on SRS improvement activity and development cost. From the statistical analysis of the data, we found the effectiveness of the PQM for risk evaluation in terms of cost overrun. We also found the effectiveness of improving the SRS by the advice which is derived from the inspection by applying the question set to the SRS. Within our knowledge, this work is the first contribution of proposing a design methodology of inspection system which enables to inspect the SRS systematically.

In future work, we plan to analyze a detailed relation between the PQC and the difference of the cost, quality and delivery-time for risk evaluation with high accuracy. We will also apply the proposed methodology to another type of development project, including embedded system, and package-based solution.

ACKNOWLEDGMENT

The authors are grateful to Dr. T. Kitani, Dr. T. Hayama, Mr. T. Kobashi, Mr. M. Hiraoka, Ms. M. Fujinuki, Mr. A. Namikawa, Dr. N. Ohsugi, and Mr. H. Yamamoto of NTT DATA Corporation for their assistance.

REFERENCES

[1] I. F. Alexander and R. Stevens, Writing Better Requirements, Addison-Wesley, 2002.

[2] B. W. Boehm, "Verifying and Validating Software Requirements and Specifications," IEEE Software, Vol. 1, No. 1, Jan/Feb. 1984, pp. 75-88.

[3] B. W. Boehm, et al., Software Cost Estimation with COCOMO II, Prentice Hall, 2000.

[4] B. H. C. Cheng and J. M. Atlee, "Research Directions in Requirements Engineering," FOSE '07, IEEE Computer Society, May 2007, pp. 285-303.

[5] D. Damian and J. Chisan, "An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management," IEEE Trans. Software Engineering, Vol. 32, No. 7, Jul. 2006, pp. 433-453.

[6] A. Davis, et al., "Identifying and Measuring Quality in a Software Requirements Specification," Proc. of the First Int'l Software Metrics Symposium, IEEE Computer Society, May 1993, pp. 141-152.

[7] C. Denger and T. Olsson, "Quality Assurance in Requirements Engineering," in Engineering and Managing Software Requirements, A. Aurum and C. Wohlin, Eds. Springer, 2005, pp. 163-185.

[8] C. Denger and F. Shull, "A Practical Approach for Quality-Driven Inspections," IEEE Software, Vol. 24, No.2, Mar./Apr. 2007, pp. 79-86.

[9] F. Fabbrini, et al., "Achieving Quality in Natural Language Requirements," Proc. of 11th Int'l Software Quality Week Conference (QW'98), May 1998, 17 pages.

[10] M. E. Fagan, "Advances in Software Inspections," IEEE Trans. Software Engineering, Vol. SE-12, No. 7, Jul. 1986, pp. 744-751.

[11] K. Fosberg and H. Mooz, "System Engineering Overview," R. Thayer and M. Dorfman (eds.), Software Requirements Engineering, IEEE, 2000, pp. 44-72.

[12] G. Fanmuy, et al., "Requirements Verification in the Industry," Proc. of CSDM 2011, Springer, 2012, pp. 145-160.

[13] G. Génova, et al., "A Framework to Measure and Improve the Quality of Textual Requirements," Requirements Engineering J., Vol. 18, No. 1, Mar. 2013, pp. 25-41.

[14] T. Gorschek and A. M. Davis, "Requirements Engineering: In Search of the Dependent Variables, Information and Software Technology, Vol. 50, No.1-2, Jan. 2008, pp. 67-75.

[15] IEEE Std. 830-1998, Recommended Practice for Software Requirements Specifications, IEEE, 1998.

[16] IEEE Std. 1028-1997, Standard for Software Reviews, IEEE, 1997.

[17] JISA, Requirements Engineering Body Of Knowledge (REBOK), Ver. 1.0, Kindaikagakusha, 2011 (In Japanese).

[18] E. Kamsties, et al., "Detecting Ambiguities in Requirements Documents Using Inspections," Proc. of WISE'01, 2001, pp. 68-80.

[19] E. Knauss, et a., "Investigating the Impact of Software Requirements Specification Quality on Project Success," Proc. of PROFES '09 LNBI Vol. 32, Jun. 2009, pp. 28-42.

[20] J. Krogstie, et al., "Towards a Deeper Understanding of Quality in Requirements Engineering," Proc. of CAiSE '95, LNCS Vol. 932, Springer, 1995, pp. 82-95.

[21] C. J. Neill and P. A. Laplante, "Requirements Engineering: The State of the Practice," IEEE Software, Vol. 20, No. 6, Nov./Dec. 2003, pp. 40-45.

[22] NTT DATA, http://www.nttdata.com/global/en/.

[23] K. Pohl, Requirements Engineering, Springer, 2010.

[24] A. Porter and L. G. Votta, "An Experiment to Assess Different Defect Detection Methods for Software Requirements Inspections," Proc. of ICSE 1994, IEEE Computer Society, pp. 103-112.

[25] S. Saito, et al., "Empirical Analysis of the Impact of Requirements Traceability Quality to the Productivity of Enterprise Applications Development," Proc. of APSEC 2012, IEEE Computer Society, Dec. 2012, pp. 330-333.

[26] S. Saito, et al., "Requirements Clinic: Third Party Inspection Methodology and Practice for Improving the Quality of Software Requirements Specifications," Proc. of RE 2013, IEEE Computer Society, Jul. 2013, pp. 290-295.

[27] F. Salger, et al., "An Integrated Quality Assurance Framework for Specifying Business Information Systems," Proc. of CAiSE Forum 2009, Jun. 2009, pp. 25-30.

[28] F. Salger, Requirements Reviews Revisited: Residual Challenges and Open Research Questions, Proc. of RE 2013, IEEE Computer Society, Jul. 2013, pp. 250-255.

[29] F. Shull, et al., "How Perspective-Based Reading Can Improve Requirements Inspections," IEEE Computer, Vol. 33, No. 7, Jul. 2000, pp. 73-79.

[30] I. Sommerville and P. Sawyer, Requirements Engineering: A Good Practice Guide, Wiley, 1997