Measuring task quality through technical-service support software

G. Takahashi*1, S. Yamada2

1. University of Tsukuba: 3-29-1 Otsuka, Bunkyo-ku, Tokyo, 112-0012, Japan
2. Keio University: 3-14-12 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, 223-8522, Japan

*tkgen@gssm.otsuka.tsukuba.ac.jp

Abstract:
The purpose of this research is to provide an effective model that can be used to improve the design quality of technical-service support software and thus enhance task quality. Herein, variables and model derived from previous studies are applied to a technical-service model that creates a direct chain of linkage from user contexts and software quality to overall task quality. In order to confirm its effectiveness, the proposed model is then applied into six types of technical-service support software.

The software development for enhancing “information believability” was found to be the most effective key to task quality enhancement. In contrast, improvements to integrity and reliability do not contribute to enhancing task quality, even though their needs, in terms of technical-service support software, cannot be neglected. The proposed model can be used to determine acceptable target quality in technical-service software development situations. Because the proposed measuring model treats engineer task recognition as user context information and can be used for engineer task-quality evaluation, it can precisely indicate not only the influence of a software quality as it relates to task quality but also the quality needs and the degree to which it needs to be validated.

Keywords
User evaluation, information quality, quality design, service development

1. Introduction

Recently, the manufacturers of industrial equipment and devices have begun working to enhance revenue by offering improved after-sale services for their products, such as maintenance and technical consulting. In the wireless-communications industry, some mobile network operators now receive technical services from equipment manufacturers after the initial network infrastructure is established. Such technical services fall into two primary types. One is “base-station deployment design”, which aims to study the relevant technological aspects and determine the proper mix of equipment such as base stations, antennas, and devices, before commercial service begins, and the other is “area investigation”, which aims to investigate service levels of operator’s current network.

In this research, the manufacturers mentioned in the paragraphs above, which are called “technical-service suppliers”, provide services related to network establishment as well as operation and maintenance instead of the customers and mobile network operators. These technical-service suppliers also deliver technical support and consulting as part of the technical-service value they provide.

The technical-service process consists primarily of “development” and “delivery”. In the development process, engineers employed by the technical-service supplier obtain the technologies necessary to create and deliver data and documents. In the delivery process, those engineers create and deliver the data and documents. Furthermore, they explain the contents of documents in details to their customers. The technical-service suppliers develop software that ensures a proper level of service quality and productivity is maintained. However, technical-service suppliers often outsource the software development. In this research, the purchased software is called “Technical-Service Support Software (TSSS)”.

After accumulating technical knowledge through the experiences, suppliers apply that knowledge to the TSSS quality enhancement process. One of the most important issues related to enhancing technical-service quality is determining how to measure task quality via the TSSS, and how to determine what software quality improvements should be pursued. In this research, task quality which means quality of performed task is obtained by determining
task fitness via a process recognized by a technical-service supplier engineer, through the TSSS, during the technical-service process.

In the wireless-communications industry, a TSSS has two main functions: design and area investigation (Umemoto et al., 2006). The design TSSS is used to formulate a base-station deployment plan and visualizes simulation data for electric power distributions and communication quality on a digital map system such as a Geographic Information System (GIS). These simulations help customers determine their investments and select solutions for technical issues. Since the design TSSS handles a large variety of input and/or output information, it requires software capable of linking with various systems such as a GIS and dedicated customer systems.

In contrast, the area investigation TSSS is used to check on quality and performance in an existing customer network. In order to grasp technical issues related to network operations, technical-service supplier engineers obtain throughput and protocol data by means of test sets and then use this information to analyze communication failures and quality deterioration.

A TSSS has several features. First, the software is designed for engineers who often must fulfill multiple roles in order to discharge their responsibilities. This can include service development, outsourcing software development, and service delivery. As a result, technical knowledge requirements are deeper and more specialized for a TSSS than for software in general. This feature is reflected in the size of the technical-service engineering department.

Second, since a technical-service is designed to deliver solutions or technical information, the value of the service is found in the information related to technical aspects that it provides via documents and consulting. Therefore, the functional quality of the service depends on the information quality. In terms of non-functional quality, the reliability of a TSSS is different from the reliability of packaged software. Furthermore, since a TSSS deals with new technology and technical issues, the service process is often volatile. Consequently, flexibility is required when fixing standard processes and confirming usability.

Even though the ISO/IEC-9126 (2001) software quality model is widely applicable to most cases, it is sometimes hard to apply it to practical TSSS situations. ISO/IEC-9126(2001) evaluates software quality and its influence by using an observation variable comprehensively. In the case of applying into the technical-service, misunderstandings on the quality that should be improved may occur when considering an evaluation structure of the task quality depending on technical-service process and contents, and taking care of an important sub-characteristic of software quality are not enough. As a result, a measuring model such as the SEM (Structural Equation Modeling) which can handle observed variables, latent variables and their relationship that govern software quality and its effects is required.

Third, a TSSS should be capable of task quality evaluations because engineers adopt TSSS features that correspond to specific purposes, such as design and area investigation. Therefore, a TSSS quality design-measuring model must be capable of separating and specifying the quality measurements necessary to develop a structured engineer’s task quality evaluation.

The purpose of this research is to provide an effective measuring quality model for TSSS that will enable users to determine specific variables of a quality design in order to enhance task quality based on the traditional variables of a technical-service business domain. The proposed measuring model is then applied to the case of a TSSS in the wireless-communications industry. The obtained results clearly show that the quality need and required improvement can be precisely analyzed using our proposed model.

2. Technical-service support software measuring model

2.1 Review of previous studies and structure of the measuring model

Numerous measuring models for system evaluation have been proposed and demonstrated to date (Adams et al., 1992; Davis, 1989; DeLone and McLean, 1992, 2003; Goodhue and Thompson, 1995; Chikara and Fujino, 1997; Seddon, 1997; Leidner, 1997; Venkatesh and Davis, 2000; Rai and Walker, 2002; Iivari, 2005; Leclercq, 2007; Venkatesh and Bala, 2008). In terms of user evaluations of information systems and software, the measuring models discussed in previous studies and our research are shown in Fig. 1. The structures of these measuring models can typically be summarized in three phases: user context, software quality, and software quality influences.
In order to grasp the context of software quality and its effect on usage, user contexts are required. For a TSSS, the weight of importance for quality will differ according to the engineer’s context, which depends on the engineer’s task recognitions, or on service development and delivery roles. Understanding the relationship between software and usage quality levels leads to a clear appreciation of the influence of software quality development on task quality via TSSS. This research, the task recognition mentioned later, is added as a user’s context.

The ISO/IEC-9126 (2001) software quality model shown in Fig. 1 specifies external and internal quality from the developer’s viewpoints. The issues considered are functionality, reliability, usability, efficiency, conservativeness, and portability. Iivari (2005) applies the external and internal quality of the ISO/IEC-9126 (2001) to the latent variables of information and system quality, respectively. Goodhue and Thompson (1995) treat latent variables which do not tell software quality and task-technology fit apart. For the influence of the software shown in Fig. 1, ISO/IEC-9126 (2001) has specified quality in use for the software quality model at the time of use from a user’s viewpoint. Quality in use relates to productivity, effectiveness, safety, and satisfaction. Iivari (2005) showed quality in use as an individual performance, and handles the relationship between quality and individual performance through usage behavior.

The latent variables integrated software qualities with task-technology fit cannot grasp the relation between software qualities and task quality. In addition, in previous studies (Goodhue and Thompson, 1995; Iivari, 2005) it was found to be difficult to specify what kind of process productivity is improved because the studies focused on individual performance results instead of the process. However, an understanding of the technical-service processes is needed in order to evaluate task quality through a TSSS because a service commodity is a series of processes that include developing element technology, producing report documents, and delivering technical consulting. Therefore, in technical service, the process of measuring task quality through the TSSS should focus on the development and delivery task processes.

In this research, first, observed variables of user context are focusing on a certain task or role as engineer’s task recognition. Second, Iivari (2005) treats latent variables about system quality and information quality and analyze correlation between the latent variables on the phase of software quality. In this research, observed variables related to system quality and information quality are used in TSSS quality in order to properly reflect design quality with taking into account of the technical-service features. Third, task quality is measured the quality of performed tasks on the phase of software quality influences, from developing a service technology to delivering a technical service. The task quality is a latent variable constructed from several observed variables through the technical-service process. This fact leads that task quality enables path coefficient changes to be expressed according to service types such as deployment design and area investigation. As a result, a measuring model works to help others comprehend an engineer’s perception of the relationship between variables. Therefore, the proposed model takes the form of SEM drawn in Fig.1.
2.2 Task recognition

The task-characteristic by Goodhue and Thompson (1995) shown in Fig. 1 is treated in the two variables of routineness and interdependence (Perrow, 1964). There are two processes in the technical-service. One is the development process, and the other is delivery process. The development process consists primarily of non-routine tasks. Because the delivery process consists of a mix of routine and non-routine tasks, the task-characteristic of engineers who engage in both types of processes cannot be clearly distinguished.

Furthermore, in terms of interdependency as a task characteristic, engineers in technical-service business divisions share technical knowledge with Research and Development (R&D) division engineers. The engineers in business development division also share budget cost information and customer’s requirements with sales accounting divisions’ members. Since the engineer in the business development division routinely arranges tasks over several divisions in technical-service projects, the task characteristic of interdependency is not adequate for the variables of the user’s context.

In order to analyze task quality through TSSS, it is necessary to take into consideration engineer task recognition regarding engineers’ roles. However, the previous studies (Ives et al., 1983; Joshi, 1990; Goodhue and Thompson, 1995; Petter et al., 2008) treat information on a division’s name or a user’s job title as a task characteristic, and analyze the influence of the task characteristic rather than the user’s context regarding roles on the task-technology fit.

Consequently, a decision based on the task characteristic such as a division’s name or a user’s job title has the potential to misrepresent a quality needed to enhance the task quality through the TSSS. In addition, information about the relationship between user context and software quality is not suitable for practical situations in a technical service. Thus, variables for task recognitions are required to define user context.

The proposed model measures the consciousness strength, regardless of what job type the engineer engages in, while simultaneously focusing on a certain task or role as task recognition. In this research, task recognition is defined as a software user’s recognition of a business role. An engineer pays attention in task recognition to two technical-service development and delivery processes. “Development task recognition” refers to the strength of the task recognition that represents an engineer’s ability to develop solutions and implement technologies. “Delivering task recognition” is the strength of the task recognition that means an engineer performs information delivery and consulting to a customer. These two variables are included in the proposed model.

2.3 Technical-service support software quality

TSSS-defined software qualities for internal and external user evaluations form the engineer’s perspective. These are functionality, reliability, usability, effectiveness, maintainability, and portability in ISO/IEC-9126 (2001). Measuring models proposed in previous studies (Goodhue and Thompson, 1995; Iivari, 2005) are difficult to apply to practical technical-service situations because those models are required to treat numerous variables and handle more than 500 samples.

In addition, a measuring variable that directly connects to design quality is preferred. Therefore, based on the abovementioned TSSS characteristics, the functionality and the reliability of ISO/IEC-9126 (2001) are specifically employed as variables. A few comments have something to do with Ease/Training” (Goodhue and Thompson, 1995) or usability (Bailey and Pearson, 1983) are received during a pilot study interview, which will be discussed in Section 3.1. Then, no correlation between usability and each variable of the task quality in later 2.4 is confirmed.

The functionality of ISO/IEC-9126 (2001) includes the “suitability”, “accuracy”, “interoperability”, and “security” sub-characteristics. TSSS functionality corresponds to the three former sub-characteristics and depends on knowledge and information in the business domain. For suitability and accuracy, providing useful information is demanded during consultations or discussions with the customer. When the information does not meet the customer’s needs, even if provided results are correct and detailed, the technical-service engineer’s task quality evaluation may suffer. It is important for output information from the TSSS to agree with the engineer’s individual experience and theoretical knowledge. Therefore, “information believability” in Kahn, Strong and Wang (2002) is included in the proposed model. “Information believability” considers customer satisfaction and determines whether the information is sufficiently accurate to persuade.

Interoperability is referred to as “compatibility” by Goodhue and Thompson (1995), where it is defined as the ability to compare and integrate data consistently. In the case of TSSS, it is important for software to possess not only compatibility but also linkage of another type of software and test sets. Therefore, the “integrity” discussed by Bailey and Pearson (1983) is included in the proposed model.

The reliability characteristic in ISO/IEC-9126 (2001) has sub-characteristics for “maturity”, “fault tolerance”, and “recoverability”. Packaged software is required to have a high grade of “maturity” and “fault tolerance” in order to guarantee its grade through proven track records or the product life cycle. However, it should be noted
that the required grade of TSSS reliability is a different characteristic from that of packaged software.

In the case of a TSSS, it is important to recover immediately and continue providing service, such as by creating a document, even if an error occurs, which is why it is often permitted lower maturity and fault tolerance levels than packaged software. Therefore, “Recoverability” is included in the proposed model to indicate recovery is immediate when an error occurs. Iivari (2005) uses recoverability as an observed variable for the latent variable called "System Quality". Because the interpretation of the variable is different, quality properties of ISO/IEC-9126 (reliability) are regarded as the name of observed variable in this research. In addition, measuring recoverability of the sub-characteristic as question contents is intended.

TSSS qualities are defined as external and internal qualities that are evaluated by users or engineers. There are three variables: “information believability”, “integrity”, and “reliability”. “Information believability” represents the credibility of output information, “integrity” represents the ease with which software data are linked with another system, and “reliability” represents how quickly the software recovers when failures and/or errors occur.

2.4 Task quality through technical-service support software

The variables for task quality take task recognitions and two technical-service processes, development and delivery, into account. In the development process, the supplier’s engineers establish the delivery process and fundamental technology for creating consulting documents. In the delivery process, the same engineers create documents and materials and make presentations to their customer. The variables for task quality through TSSS treat task accomplishment enhancement through a series of technical-service processes.

Previous studies ISO/IEC-9126 (2001), Goodhue and Thompson (1995), and Iivari (2005) treated software users or system operators as customers, and did not define users as suppliers. For this research, the observed variables for task quality take the relationship between the engineer users as suppliers and engineer’s customers into account.

Because a technical service is regarded as a professional service, like a medical or legal service, the technical service refers to three perspectives that are often applied to a medical service: expert skill quality, physical quality, and interaction quality discussed in Fujimura (1995). TSSS task quality related to development tasks corresponds to expert skill quality and the “trial” portion of the development process. TSSS task quality in relation to delivery tasks corresponds to the physical and interaction qualities. The physical quality is regarded as the quality of the document, called “information delivery”. Interaction quality is regarded as the quality of the discussion with the customer based on TSSS output information, and is called “customer interaction”.

Therefore, in this research, the task quality achieved through the TSSS is treated as the fitness with which engineers in technical-service suppliers recognize tasks through technical-service processes. The TSSS task quality is constructed in the three variables of “trial”, “information delivery”, and “customer interaction”. “Trial” relates to how easy or difficult it is to conduct various trials related to problem solution. “Information delivery” explores whether the quality of the documents that engineers can provide increases during technical-service delivery. “Customer interaction” explores whether the quality of arguments with the customer increases in service delivery.

2.5 Proposed measuring model of task quality

Finally, by adding the variables for task recognitions as user context, the proposed measuring model, which follows the structure and variables described in 2.1 through 2.4, is shown in Fig. 2. The proposed measuring model has a chain relationship between task recognitions, TSSS quality, and task quality.

The proposed model shown in Fig. 2 has two primary advantages when compared with conventional models (ISO/IEC-9126, 2001; Goodhue and Thompson, 1995; Iivari, 2005). One of the advantages is the ability to measure software quality with respect to user context. This fact makes it possible to understand sufficiency or deficiency levels based on the measurement results of task recognition strengths such as “development task recognition” and “delivery task recognition”. Furthermore, each task-recognition function provides user context information related to a software quality analysis influence on task quality.

The other advantage is its ability to analyze the relationship between software and task quality constructed variables with regards to path changes in the proposed model. Additionally, provisioning specific variables for the task quality takes the evaluation structure into account and applying the proposed measuring model enables us to analyze software quality needs and determine which quality element contributes to improving task quality based on the technical-service contents. Furthermore, analyzing the influence of software quality on task quality makes the technical-service process improved clearly specified owing to the path changes from task quality.
3. Model application case study

3.1 Data collection in the wireless-communications industry

In this case study, a data set is obtained from the technical-service engineers in a manufacturing company that engaged in the development and delivery of a technical service for a network service business planning division. A questionnaire-based pilot study is conducted and the data replies received from 16 engineers are analyzed. The terminology used for each variable in the questionnaire is decided based on the discussion with the engineers in order to ensure that the content was reported properly. The items about the task recognition and the task quality are not treated in the previous studies. Next, three of the technical-service engineers that participate in the pilot study are interviewed. Their responses allow us to confirm that each variable in this research is consistent with the technical-service. The observed variables in the questionnaire are shown in Appendix A.1. A seven-point Likert scale (“Strongly agree” to “Strongly disagree”) is adopted.

Based on the pilot study results, we then conduct a questionnaire survey of 29 engineers on six TSSS types. Since each engineer’s responses cover one or more of the TSSS types, 85 responses are obtained, as shown in Table 1. The technical-service supplier engineers are engaged in both service development and software procurement tasks. In fact, two of them are engaged in all task roles. The number of responses for each software type is tabulated in Table 1.

Table 1: Software type and the total number of responses

<table>
<thead>
<tr>
<th>Software type</th>
<th>Purpose of the software</th>
<th>Number of responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software 1</td>
<td>Outdoor Design</td>
<td>27</td>
</tr>
<tr>
<td>Software 2</td>
<td>Indoor Design</td>
<td>13</td>
</tr>
<tr>
<td>Software 3</td>
<td>Indoor Design</td>
<td>8</td>
</tr>
<tr>
<td>Software 4</td>
<td>Area investigation</td>
<td>19</td>
</tr>
<tr>
<td>Software 5</td>
<td>Outdoor Area investigation</td>
<td>9</td>
</tr>
<tr>
<td>Software 6</td>
<td>Indoor Area investigation</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>85</td>
</tr>
</tbody>
</table>

3.2 Six TSSS cases

3.2.1 Relationship between task recognition and TSSS quality

Six cases involving a technical service in the wireless-communications industry were analyzed by SEM, and the results are shown in Fig. 3. Note that, the errors of variables for TSSS quality, task quality, and task quality through TSSS are omitted in Fig. 3. In this analysis, each path coefficient is the value for most likely estimation,
and the errors variances are presumed to be constant values. According to a tutorial for SEM (Asano et al., 2005), the obtained results will ensure a moderate level of fitness. The value of CFI is satisfied with more than 0.95 though the value of NFI is slightly less than 0.9. This level of NFI may be due to small samples size. The proposed model ensures moderate level of fitness even though the proposed model still has an issue of validity depending on sample size. Therefore, the proposed model measuring model in Fig.2 is deemed to be acceptable for the six cases.

The results show that “delivery task recognition” negatively influences “reliability”. This fact implies that as “delivery task recognition” is increased, engineers are experiencing a lack of reliability in the TSSS. In the same manner, “development task recognition” negatively influences “integrity”. Even though delivery task recognition interactively influences “development task recognition”, “delivery task recognition” does not influence “integrity” and “development task recognition” does not influence “reliability”.

“Development task recognition” might have a positive influence in cases where the user is a developer, which is why developers are likely to evaluate the software quality leniently. As shown in Fig. 3, engineers evaluate TSSS quality from outsource provider, not developer, viewpoints. However, it is possible that an engineer who outsources software development to a technical-service supplier might evaluate software characteristics more strictly than another engineer because engineers who outsource software development are seldom lenient when evaluating the software quality of the results. Therefore, as can be seen in Fig. 3, “development task recognition” does not positively influence all TSSS qualities.

These results are consistent with Goodhue and Thompson (1995). The relationship between “task characteristics” and “task-technology fit” indicates a negative influence on “reliability” or “compatibility”. In addition to these findings, it was also determined that “delivery task recognition” has no influence on “integrity” or “reliability”. Therefore, the proposed measuring model enables us to indicate the requirement changes received from engineers in technical-service processes, and to precisely observe what kind of quality is lacking due to task recognition treatment.

3.2.2 Relationship between TSSS quality and task quality

The results obtained indicate that improvements to “information believability” are most effective for improving TSSS “task quality”. This fact implies that engineers are sensitive to information consistency with physical principals such as radio propagation theory, and that the engineer’s technical experience or knowledge of network protocol is a relevant factor. Sufficient levels of consistency lend persuasiveness to both supplier and customer engineers. No significant influence of each variable for task recognitions on the “information believability” in Fig.3 unclearly indicates insufficiency of the “information believability”. In addition, although “reliability” was found to be related to “delivery task recognition”, it had no influence on “task quality”. It should be noted that
recognizing when quality is insufficient does not always contribute to enhancing “task quality”.

The proposed measuring model is different from Iivari’s model in that, where Iivari’s model has correlations between several quality-factors, the proposed model employs an observed variable in order to specify a task quality influencers. It has been observed that treating a latent variable constructed the observed variables is useful for finding correlations, but it is difficult to specify a direct cause related to task quality enhancement through the TSSS. As previously mentioned, “information believability” is the most effective tool for enhancing technical-service “task quality” through the TSSS. Therefore, observed variables should be added to the measuring model if the proposed model directly specifies each variable and its coefficient.

“Customer interaction” indicates the largest coefficient value. This is due to the fact that enhancing customer interaction through meaningful discussions on development trials is more valuable than “information delivery” via reporting documents. When “task quality” influences “customer interaction”, engineers regard the effect of technical service through the TSSS as a valuable factor. In contrast, “trial” only has a slight influence on “task quality”, which is why engineer evaluations in a technical-service supplier recognizes “trial” as an indirect process for technical-service performance. Therefore, the findings in Fig. 3 represent the task quality constructed variables associated with a service encounter, such as a technical-service delivery process.

3.3 Comparing by software purposes

3.3.1 Technical-service support software to design base station deployment

Cases where TSSSs were used to design base station deployments are analyzed in Fig. 4. In comparison with the TSSS used for area investigation (discussed in the following section), the TSSS used for network design has the following features. First, “development task recognition” has a stronger negative influence on “integrity”. In the case of the TSSS used for a design purpose, this is considered to be due to the fact that technical service engineers become more sensitive about the inability to integrate insufficient quality.

Second, “integrity” has a stronger influence on task quality. In this case, the influence of “integrity” refers to how it has become easier to link with other systems because numerous input-output interfaces are provided. This is one of the features of a TSSS used for design.

Third, “task quality” in terms of “customer interaction” has a stronger affect in terms of “information delivery”. When a design proposal is delivered to customers, it is important for engineers in technical-service suppliers to prove that the quality level of document and consulting are in keeping with the engineer’s expertise.

![Fig. 4: TSSS for design cases](image-url)
3.3.2 Technical-service support Software used for area investigation

The TSSSs used for area investigation are analyzed in Fig. 5. A TSSS used for area investigation has the following features, which differentiate them from the TSSSs used for design that were discussed in the previous section. First, “delivery task recognition” has a stronger negative influence on “reliability”. As a result, in cases related to area investigation, it is important for engineers to ensure reliability in order to satisfy requirements and complete document creation. This means that engineers are likely to perceive quality as more insufficient.

Second, “information believability” is much more effective for improving “task quality” on the TSSS. Excess efforts at enhancing reliability to the point where the quality level is improved more than enough to compensate for insufficiency does not contribute to enhanced task quality. Therefore, it should be noted that even when the software quality is recognized as insufficient, it is not always appropriate to focus on enhancing task quality.

Third, the dominant portion of the “task quality” is “information delivery”. This fact relates to a difference between the area investigation and design service types. The value of a technical service for area investigation is found in the numerical data themselves. In contrast, the value of a design technical-service is extracted from experience obtained thorough consulting, along with the report documentation and technical information gathered by means of the TSSS. In other words, the difference depends on the characteristics of the technical-service contents.

Thus, for area investigation, enhancing “task quality” is perceived by improved delivery of exact numerical data and report documents. In contrast, for design, enhancing “task quality” is perceived via the engineer’s technical consulting efficacy. Therefore, the task quality evaluation is reflected in the service commodity characteristics by means of the proposed variables. Which path coefficient from the software quality toward task quality is the largest is clearly observed.

3.4 Comparing by software purposes

Applying the proposed measuring model shown in Fig. 2 into the case of wireless communication industry, the results are obtained in Fig.3 to Fig.5. Thorough our analysis in Fig.3 to Fig.5 based on the proposed model Fig.2, it was determined that application of the measuring model provides the following useful information for software quality design and the planning of software products. First, the proposed model can be applied to determine adequate target quality. More specifically, the model enables users to simultaneously consider qualities for enhancing task quality and reducing quality related complaints. In the case of the TSSS for design, the quality that an engineer tends to regard as insufficient contributes to enhancing the task quality. Conversely, qualities enhancing task quality are different from qualities that engineers tend to regard as insufficient in the case of a TSSS for area investigation.

These findings lead to the conclusion that remarkably enhancing “integrity” is expected to enhance task quality. However, it was found that excess enhancement of “reliability” is not effective for enhancing task quality.
Therefore, application of the proposed model enables users to precisely determine an adequate degree of target quality.

Second, the proposed model indicates that improvements to “information believability” are the most effective quality for improving “task quality”, which makes “information believability” a fundamental TSSS requirement. The development for enhancing “information believability” is effective for improving the “task quality” in both of TSSS purposes, design and area investigation. Furthermore, according to the results in TSSS design cases, it should be noted that simultaneously enhancing “integrity” not only helped evaluate poor quality, it also enhanced task quality. Furthermore, in the case of the software used in area investigation, ensuring “reliability” enough to be acceptable and enhancing “information believability” are also effective for improving “task quality”.

Third, the proposed model is a tool for classifying technical-service types depending on the task quality evaluation structure. Task recognition results indicate which process, development or delivery, is more highly weighted in terms of importance, and which service commodity is more valued, such as consulting experience in the network-design process or the data product in the area investigation process. These features are derived from path coefficients that aim toward “task quality”.

4. Conclusion

This research proposes a measuring model of task quality through TSSS that can be used to enhance quality design. The proposed model has two features: treating added task recognition as a variable that grasps the user TSSS context, and providing variables for the concept of the task quality while focusing the process of performing a technical-service task, and structurally understanding the concept of the task quality through engineering perceptions.

When the proposed model is applied to a TSSS in a wireless-communications industry and a technical service for design and area investigation, it was found that it could specify variables and influence relationships between task recognition, software quality, and task quality in the technical-service context. Furthermore, it was determined that improving the quality perceived to be insufficient, depending on task recognitions, does not always enhance task quality, and that enhancing “information believability”, which is different from the quality perceived to be insufficient, contributes to enhancing task quality. These findings provide useful knowledge for software engineers and software development planners desiring to enhance TSSS quality design.

In this research, we applied the model to a wireless-communications industry example. When the proposed model is applied to another professional service, verification using specific quality software will be necessary. Although existing software was used in this study, it will be worthwhile to confirm the variables via a more suitable TSSS in the future. Furthermore, additional considerations will be necessary before the model can be applied to a small organization with limited amounts of user sample data.

Task recognition and the task quality variables focus on the technical-service process types. Developing variables that focus on the breakdown processes are useful. The proposed model is capable of handling consecutive relationships between task recognition, TSSS quality, and task quality. It was also determined that an adequate target quality level can be precisely analyzed by gaining an understanding of these consecutive relationships.

References:


**Acknowledgements:**

The authors sincerely thank the reviewers for their constructive comments. This work was supported by JSPS KAKENHI Grant Number 26350414.

**Author’s biographical notes**

Gen Takahashi is a doctoral student at Graduate School of Business Sciences, University of Tsukuba. His research interests encompass flexibility strategy, service quality management and system science in a technical-service.

Dr. Shu Yamada is a professor, Faculty of Science and Technology, Keio University. He teaches applied statistics and quality management at undergraduate and graduate schools. His research interests focus on total quality management, design of experiments for quality, statistical quality control and so forth.

[DOI : 10.17929/tqs.3.22]  
Received: March 8, 2016  
Revised: November 30, 2016  
Accepted: April 19, 2017
Appendix:

A.1: Questionnaire

A seven-point Likert scale was adopted except user’s attribute data. Alternatives are omitted except user’s attribute data. The “Software X” refers to “Software 1”, “Software 2”, ..., “Software 6”. In our pilot study, “usability” is confirmed by the measure based on Bailey and Pearson (1983).

<User’s attribute data>
Please provide your technical-service support software experience level and your job title. Choose the most appropriate number from the alternatives.

<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
<th>Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What is your job title?</td>
<td>1. Manager/Expert, 2. Assistant Manager, 3. Staff</td>
</tr>
<tr>
<td>2</td>
<td>How long have you used Software x?</td>
<td>0. Not at all, 1. Less than 1 month, 2. 1 month to 3 months, 3. 3 months to 6 months, 4. 6 months to 1 year, 5. More than 1 year</td>
</tr>
</tbody>
</table>

<Task recognition>
Please describe your role and actual usage. Choose the most appropriate number from the alternatives.

<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>You develop new element technology and/or solution business in a technical service. (Development task recognition)</td>
</tr>
<tr>
<td>2</td>
<td>You conduct meetings to discuss technical issues with customers. (Delivery task recognition)</td>
</tr>
</tbody>
</table>

<Quality of technical-service support software>
Please provide your opinion on the quality of Software X. Choose the most appropriate number from the alternatives.

<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Software X is easy to link with other software and hardware systems (Integrity)</td>
</tr>
<tr>
<td>2</td>
<td>Software X error recovery is fast. (Reliability)</td>
</tr>
<tr>
<td>3</td>
<td>Software X is easy to use. (Usability)</td>
</tr>
<tr>
<td>4</td>
<td>Software X output information is so credible that engineers can explain logically to their customer. (Information believability)</td>
</tr>
</tbody>
</table>

<Task quality through Soft X>
Please describe your opinions about Software X’s suitability for use. Choose the most appropriate number from the alternatives.

<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Software X contributes to making various problem-solving trials (Trial)</td>
</tr>
<tr>
<td>2</td>
<td>Software X contributes to enhancing the level of deliverable information quality (Information delivery)</td>
</tr>
<tr>
<td>3</td>
<td>Software X contributes to enhancing discussion levels with customers related to technical issues (Customer interaction)</td>
</tr>
</tbody>
</table>
### A.2: Correlation between observed variables

Each correlation coefficient between observed variables is tabulated in Table A.2.

<table>
<thead>
<tr>
<th>No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Development task recognition</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Delivery task recognition</td>
<td>0.46</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Integrity</td>
<td>-0.28</td>
<td>-0.09</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Reliability</td>
<td>-0.14</td>
<td>-0.26</td>
<td>0.17</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Information believability</td>
<td>0.10</td>
<td>-0.03</td>
<td>0.03</td>
<td>0.01</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Usability</td>
<td>0.00</td>
<td>0.04</td>
<td>0.29</td>
<td>0.26</td>
<td>0.17</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Trial</td>
<td>-0.16</td>
<td>0.03</td>
<td>0.10</td>
<td>0.20</td>
<td>0.10</td>
<td>0.15</td>
<td>1.00</td>
</tr>
<tr>
<td>8.</td>
<td>Information delivery</td>
<td>0.12</td>
<td>0.11</td>
<td>0.13</td>
<td>-0.05</td>
<td>0.30</td>
<td>0.08</td>
<td>0.47</td>
</tr>
<tr>
<td>9.</td>
<td>Customer interaction</td>
<td>0.13</td>
<td>0.09</td>
<td>0.09</td>
<td>0.02</td>
<td>0.33</td>
<td>-0.02</td>
<td>0.41</td>
</tr>
</tbody>
</table>