A Survey of Communication Content in Software Requirements Elicitation involving Customer and Developer
Authors

Noraini Che Pa$^1$
and Abdullah Mohd Zain$^2$
Department of Information System, Faculty of Computer Science and Information System, Universiti Putra Malaysia, Malaysia
²Programming and Software Technology Research Group, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia
Abstract

At the heart of software requirements elicitation lies the communication between customer and
developer. There are several valuable components of communication such as medium, sender, receiver, and messages, which
relates to the input and output from both parties. Most of these messages are delivered through incompletely, inconsistently or
inaccurately defined communication medium. This study has been done to look into the communication content of the current communication
practices between developer and customer in Malaysia. The results of this study revealed some important notes on the practices of communication
content during software requirements elicitation process in Malaysia.
Keywords: requirements elicitation; software requirements specification; communication content.
Introduction

In general, organization is complex, hence identifying the requirements are especially difficult. In
addition, software requirements always change from time to time. Requirements elicitation involves the communication process
between customer and developer during the analysis phase in software engineering. There are several important components under
consideration during communication, such as the medium, sender, receiver, and the content of messages, which relates to the input and output from
both parties. Such information by the customer, which is often delivered verbally and not in writing, will be used to produce Software
Requirements Specification document (SRS). At present, several studies have been conducted on the practices of requirements elicitation but none has
looked into the communication content between customer and developer.
In practice, communication activity involves messages transmission from sender to receiver, whereby the discussion topic revolves around domain application.
(Drake et al 1993), business requirements, system barrier and others problems (Paetsch et al. 2003). While messages are in the form of
information and knowledge, knowledge is difficult to transmit because it belongs to a person who manages the particular knowledge.
According to Stary (2002), knowledge of an organisation covers tasks and processes that are carried out by customers. Such information and
knowledge are in turn used to produce the software requirements document, which is traditionally viewed as a document that communicates the
requirements of the customer to the developer who is responsible to build the system. The collection of requirements and its representation must be
understandable by both customer and developer.

The remainder of this paper is organized as follows. Section two will
describe in detail the software requirements elicitation process and the related works. Section three will present the survey results from the
requirements elicitation between customer and developer as practiced in Malaysia. Finally, Section four will conclude the
findings with some indications for future work.
According to Coulin et al (2005), requirements
elicitation is the process of searching, revealing, acquiring, and detailing of requirements for computer-based systems. This process is complex as
it involves various activities, techniques, approaches, and support tools. More often, these processes are carried out repeatedly (Aurum &
Wohlin 2005). Requirements elicitation is also looked as a negotiation process among stakeholders in order to achieve an agreement on
the system to be developed. Sommerville (2001) identifies activities involved during requirements elicitation as discovered, negotiation,
and documentation. According to Haywood and Dart (1996), these activities may be implemented using bottom-up or top-down approach, based on specific
customer problem. Aurum and Wohlin (2005) state that in general, the processes are made up by four principle activities, which are communication,
set priorities, negotiation and cooperation with the stakeholder.

Various techniques have been used for requirements
elicitation such as interviews, document analysis, group work, ethnography, prototyping, questionnaires, scenarios, and viewpoint. These
techniques may be divided into two categories: the interaction between an individual and the interaction between groups (Duran et al 2004).
Interactions between individuals are divided into two types: local and distributed. Local interaction includes prototype, group meetings,
and interviews. Whereas distributed interaction involve interaction of interviews, conferences, and meetings through video. Non-personal
interaction consists of observation, document analysis and questionnaires. According to Coulin et al (2005), most of these techniques are
adapted from various disciplines such as social science and engineering.

Requirements elicitation techniques may also be
classified into traditional, group, formal, semi formal, and natural language. In traditional ways, requirements elicitation process are performed face
to face such as through interviews, whether individually or in a group among customer or manager. There have been several difficulties
conducting interview session such as:

(i) it is time consuming;
(ii) there may exist conflict between user and manager with regards of perception, assumption, problem defined, and even
objective of a system and

(iii) different personalities and behavior (Bahn 1995), as well as
background and terminology used during communication between both parties (Liou & Chen 1993).
Nonetheless, this technique requires direct interaction between both parties; the interviewer and the respondent, which results in quick information
exchange. The quality of the information obtained is closely related to the skills of the interviewer. Basically, there are three forms of interview, which
are unstructured, structured, and semi-structured. Unstructured interviews give the respondent the freedom to express opinions, feelings,
position, goals, and beliefs of an issue. This form can be used if the interviewer has little knowledge of the domain. The weakness of unstructured interview is
the tendency of both parties to focus discussion on only specific topics. A structured form of interview allows the parties to involve and determine
the topic in advance. The results from structured interviews are easily analyzed, the process only takes a considerable short time, and best carried out
by a new analyst. However, interviewing techniques actually involve high costs and time consuming to prepare the interviews, performing the interviews
and analyzing the results of the interview. In some situations, an interview has to be conducted over time and involve several individuals, with different
needs and requirements. Finding by Hickey et al (1999) reveals that this technique is not efficient if the number of respondent
involves the public and consists of different groups.

Document analysis technique is conducted by reviewing documents and
application of an existing system. This technique is most suitable for the renovation of obsolete systems or by a new analyst. The documents
involved include design documents, manual systems, as well as forms and files used in the business processes. However, more often the
documents involved contain outdated or incomplete, and inconsistent with the current business
requirements (Hoffer et al. 2008).

Elicitation techniques that involve public participation or occur at the same time
for instance meetings, focus groups, and workshops require a designated working group. Hickey et al (1999) and Drake et al (1993) have categorized
meeting techniques that involve time and high cost as it requires the involvement of many parties at one time. Focus group is one of the
techniques performed in a group interview. This technique involves participation of the customer representatives and the developer to
exchange information through discussions (Sommerville 2007). A facilitator will be appointed to ensure that the discussions are conducted
smoothly, hence the technique is less suitable for requirement specifications of complex software systems. Meanwhile, workshop is
conducted in collaboration consisting of five stages of development, critique, understanding and support, implementation, and delay (Gottesdiener 2003),
whereby all participants play a role in every stage of the workshop conducted. This technique is able to produce high quality
requirements within a short time.

Prototyping is another requirements elicitation technique that allows user
feedback and considers in-depth information, which is considered the most suitable technique for developing the user interface requirements that
have not been identified in full. The prototype responds better to uncertain or changing requirements (Satzinger et al 2002). Two prototype
approaches are incremental and throw away. Incremental prototype is the prototype that is built in a small module from the overall user requirements.
Unlike incremental, throwaway prototyping does not preserve the prototype that has been developed. There is never any intention to convert the
prototype into a working system (Hoffer et al 2008). This technique is used to encourage user to participate in developing the customer requirements
and benefits the discussions with customers because it involves a system that is already in existence.
Meanwhile, elicitation through questionnaire requires a clear focus to ensure the information obtained is appropriate. Questionnaires are used to
gather information when the project involves many respondents and is to be completed within a short time period. The information obtained is
usually lack in depth, less authentic, and less interactive. Normally, this technique is best used to obtain information on attitudes, beliefs, and basic
features for a system. Other than questionnaire, observations may be performed by observing how users work out the actual business process
without their intervention. This technique involves high costs and requires skill to interpret and understand human actions. Often, users tend to change how they
work after finding out that they are being observed. In addition, interpretation of the observations made by the analyst is subject to influence and personal bias.
Scenario-based elicitation technique is basically a summarized description of the system as described in the beginning of the process, along the process,
and at the end of the process. The scenario is served in the form of a story and contains information on the process, actions and interactions of
users with the system. However, this technique does not show the internal structure of a system although it may be used to
understand and to validate the requirements.

The most commonly used communication type during requirements elicitation
processes are verbal, written, and mediator (Saiedian & Dale 2002, Coughlan et al 2003). The medium chosen is important to assure the
types of messages received are similar to the actual messages that were delivered. Usually, the chosen method is in favors of communication with fast
feedback time, clear, no conflict, and easy to understand. Many customers and developers alike use natural language to communicate during
requirements elicitation process. However, this method poses some problems such as differences in pronunciation, expression,
Communication Content among Software Developers in Malaysia

The general objective of this survey is to identify
communication content that relates to requirements elicitation activities between customer and developer specifically in Malaysia. The
questionnaire encompasses questions on communication content and the appropriate tools used to support the elicitation activities.
The specific objectives of this study are:

(1) to determine the input and output of requirements elicitation process and
(2) to recognize the actual processes involved during requirements elicitation. To achieve the above objectives, the following are some research
questions that need to be addressed:

1. What is the source of requirements elicitation for
communicating requirements during requirements elicitation in Malaysia?
2. What are the method and support tools used in preparing for software requirements specification document?
3. What are the roles of users’ involvement when performing requirements elicitation?
Stakeholder Background

The methods of data collection in this survey are through postal, e-mail, and interviews. The
respondents involved are software developers from various sectors in Malaysia. Questionnaires are appropriate because our data collection involves
public respondents where the distribution of the respondents is scattered. The selection of respondents is determined based on their position and
experience in requirements elicitation activity during system development. Participations came from various agencies that are categorized as government,
semi-government, private agencies with Multimedia Super Corridor (MSC) status and without. Table 1 shows the background of
the respondents who participated in this study.
Table 1: Background Respondent Selection

Please see Table 1 in full PDF version
In the following sections, we will present the analyses performed on the information gathered from 42 responses. The results of
the survey were then analyzed using SPSS.

Results

Table 2 shows the content and criteria investigated in
the survey. There are 5 categories of content, which are the requirements sources, analysis and modeling, prototype, SRS, and user involvement.
Table 2: Content and Criteria Investigated

Please see table 2 in full PDF version
Requirements Sources

Requirements sources are information that are gathered from the customers. These refer to
customer needs for new implementations or even upgrades. From the analysis, it is found that numerous sources from customers were used in
process identification requirements. The survey result shows 69.0% of respondents chose the work process as their main information source to
identify the software requirements. Other sources used are based from existing system (50.0%), 50.0% from the organization rules, 50.0%
from expert knowledge, 42.9% from documents, and 4.8% from others source (refer to Table 3).
Table 3: Sources of Requirements

Please see Table 3 in full PDF version
Many organizations choose and modify their requirements sources in accordance with technology changes. Besides, sources of project are also influenced
by changes of external factors such as economic, politic, social, regulations, financial, psychology, history, and geography. For example, an organization
that practices a bureaucratic system often faces difficulty in gathering requirements as compared to other non-bureaucratic organizations. Changes of
management and political pattern in an organization also influence in delivering the requirements sources. Such new changes may cause customer to feel
unhappy and unable to accept. Nonetheless, changes in requirements and scope will rarely affect the information delivered
as delivered through email, telephone or interview.
Analysis and Modeling Requirements

This process includes refining and modeling the requirements. From
the analysis, (see Fig. 1) the results of the study show that respondents prefer to use Structured System Analysis and Design Method
(SSADM) as compared to Object Oriented Analysis (OOA) with small percentage of preference on internal methodology. This is
probably because the traditional method is easy to understand and represent the actual customer requirements. The survey shows that
although 71.4% of developers do not use any specific software to analyze and model the requirements, 28.6% of them have considered
using the Rational Rose, Enterprise Architect or Microsoft Visio.
Fig 1. Methodology Used for Software Requirements Analysis and Modeling

Please see Figure 1 in full PDF version
Prototype

Normally in practice, a tool is used to get feedbacks on software requirements as specified by the developer.
This type of feedback is used to examine and guarantee the consistency, completeness, reality and accuracy of software requirements. According to
Sommerville (2001), this includes checking the requirements document.

From the analysis, it is shown that 71.4% of
developers used prototype techniques to validate their requirements and 28.6% chose other techniques (refer Table 4).
Table 4: Techniques of Prototype

Please see Table 4 in full PDF version
Feedback from respondents who used prototype is 30 from 42 persons, whereby prototype parts involve user interface, schedule, process flow, and work
system. Table 5 shows the use of prototype techniques to communicate system requirement that were developed in effort to seek feedback from customer.
Implementation of the prototype involves the programming language and specified software. From the analysis, it is shown that 86.7% of developers
used programming language to implement the prototype but 13.3% chose Macromedia Dreamweaver, Microsoft Visio or Microsoft PowerPoint.
Also, most respondents stated that they used combination of requirements part to show the prototype. Study found out as much as 86.7%
presented their prototype for interface, 26.7% for schedule, 80% for process flow, and 3.3% for working system.
Table 5: Parts of Requirements that Demonstrate in a Prototype

Please see Table 5 in full PDF version
Because software requirements are often
seen as abstract statements of the services provided or the constraints of a system, they are defined in various ways. Software requirements document
can also be viewed as a detailed statement that defines the process using formal mathematics of a functional system. According to IEEE (Yang &
Tang 2003), SRS documentation is a term referring to software requirements with:
(i) the capacity required by users to solve a problem or to achieve certain objectives,
(ii) the ability of the system to fulfill the contract, standards, specifications or other and
(iii) a document that reflects the ability to satisfy objective (i) and (ii). Chirinos et al (2004) report that there is actually no consensus
on the meaning of software requirements.

is described, which are through:

(i) activities,

(ii) input and output,
(iii) data definition, and
(iv) processing requirements.
In subsequent research, Gregoriades et al. (2004) define software requirements as goals to be achieved and consider the implementation through
software operating processes, machines, and humans. Software requirements are divided into two types, which are the functional
requirements and non-functional requirements. Functional requirements refer to the functions or services provided by the system. This requirement
highly depends on the software, potential users, and the type of systems. It is also known as the behavior of the system (Chirinos et al 2004).
Meanwhile, non-functional requirements refer to the constraints of the system (Paetsch et al 2003). The process of documenting the software
requirements includes activities such as creating the software requirements specifications (SRS), reviewing the SRS content, and checking the resulting
SRS. These activities are carried out to ensure the document that is created adheres to the quality standard and satisfies the customer. Basically,
software requirements document is a group of statements that needs to be written by developer (Sommerville 2001). The details of software
requirements document depends on the kind of system to be developed and the software development process (Sommerville 2001). There are various
standards in existence for requirements document such as the IEEE, ISO 9000, and others. Basic issues in IEEE standard 830-1998
pertaining the SRS document include:

1. Functionality
   What is the software supposed to do?
2. External interfaces
How does the software interact with people, the system’s hardware, other hardware, and software?
3. Performance
What are the functions of speed, availability, response time and recovery time of various software, etc?
4. Attributes
   What are the portability, correctness, maintainability, security issues under consideration?
5. Design constraints imposed on an implementation
Are there any required standards in effect, implementation
language, policies for database integrity, resource limits, operating environment?
The survey results show that respondents did follow some standard in preparing SRS documentation, among which are from the Institute of Electrical and
Electronics Engineers (IEEE), International Standards Organization (ISO) 9000-3, National Standards or internal organization. Analysis of
data showed that 53% respondent follows their own organization standard or at least refer to similar organization in writing the SRS document. While 28%
of respondents do not adopt any formal standard, 13% of respondents adhered to standard set by IEEE, 3% adhered to ISO standard 9000-3, while the
remaining 3% adhered to the National Standards.

Further analysis reveals that most of the SRS document content includes
the following items:

• Introduction

• Content
• Project background

• System cope and business

• System summary
• Interface
• Output and input
• Process
• Procedure

Meanwhile, only a small number of organizations incorporated the following additional items:
• Change control
• Storage data
• Review
Validation

As for the tools, software that is used to prepare the SRS document is mainly word processor or specific
software. Findings show that 90.5% respondents used word processor to write SRS and 7.1% use other specific software, while the remaining 2.4%
use both types of software. Examples of specific software are Microsoft Visio, Microsoft Excel and Microsoft Project.
**User Involvement**

Findings from the survey show that most customers are involved in checking the SRS document. Analysis of
data shows that 88.1 % respondent claimed customer involvement in checking on SRS document while 11.9% claimed
otherwise (indicated in Table 6).
Table 6: User Involvement

Please see Table 6 in full PDF version
Table 7 shows the itemized content of SRS document that are validated by customer. This information is gained after the
respondents were requested to list the section of SRS document that requires confirmation by customers. Analysis of
data shows that 89.2% respondents claimed involvement of customer in functional part, 73% in system scope and business
part, 73% in interface part, 73% in input and output part, and 73% other parts. All respondents state that they do not use any
specific software to check the SRS document.
Table 7: Parts of SRS that Validate by User

Please see Table 7 in full PDF version
Consolidation of the Result

Based on the survey findings reported in Section 3, the content of
communication between the customer and developer during requirements elicitation are investigated in effort to further understand the
common practices during the elicitation process. While previous researchers look for technique and sources that is used to generate SRS, there is also
researcher that focuses on support tools to facilitate communication between customer and developer during the requirements elicitation process. While
previous studies only look into user involvement for requirements validation, this study includes source of communication, user involvement and support
tool that are used in performing requirements elicitation.

Overall, the survey conducted is able to
provide insights on current communication practices during requirements elicitation activity among software developers in Malaysia. The sources for
generating the software requirements were identified by this study. The study also showed that software developers do not use any specific tools to
support all activities for requirements during the requirements elicitation process. Survey also shows that there is no specific methodology adopted by
the developers to implement the requirements elicitation process. In addition, it is found only a handful of developers who use tools to
support requirements elicitation.
Conclusion and Future Research

This paper discusses communication content between the customer and
developer during requirements elicitation process in preparing the Software Requirement Specification (SRS) document. The findings
show that most developers do not use any support tool in implementing activities during the requirements elicitation process nor do they follow any
methodology to perform requirement elicitation. Requirement document is important because it is always taken as the basis for software development,
hence a software tool is needed in creating the software requirements document.
One obvious limitation of this study is the use of only one set of questionnaire to be distributed to the developers. In this case, the information gathered is
limited to the questions asked. More in-depth information and deeper understanding may be gained if other research methods are used in
combination such as focus group and interview. Our future work intend to increase the number of participating companies and to use additional data.
gathering techniques with the objectives of getting wider and more accurate representation of requirements elicitation
practices among industrial practitioners in Malaysia.

There are other interesting issues in communication for requirements to be
explored. The issues include medium, personalities, procedures, and communication skill. At the end, our main aim in this endeavor is to facilitate
customer and developer to consciously manage future communication during requirements elicitation by looking in-depth of considering the
communication content. Effective and clear communication will produce the best software requirement documents,
which in turn will produce good software.
References

Verlag Berlin Heidelberg. Germany.

Subcultures Perspective,” Proceeding of ACM SIGCPR Conference on Supporting teams, groups, and learning inside and outside the IS function reinventing, 72-80.

Requirements Elicitation: A Content Analysis of Stakeholder Experiences,” Information and Software Technology 45, 525-537.

Study of Requirement Analysis Where End Users Take an Active Role,” Proceeding of 15th International Conference
on Software Engineering, 177-186.

Engineering Conference, 131-138.

Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, 308-313.
Software Technology 42(6), 419-428.


for Contextual User Interface Development,”
Behaviour & Information Technology 21(6), 425-440.