

INCOSE Guide for Writing Requirements. Translation experience, adaptation perspectives

Efremov, Alexander Alexandrovich
Engineer (specialist)

Senior Consultant, MHP Management- und IT-Beratung GmbH
Frühlingstraße 6, Oberasbach 90522 Germany
Alexander.Efremoff@gmail.com

Gaydamaka, Kirill Igorevich
Graduate Assistant

MIREA - Russian Technological University
119454 Moscow, Prospekt Vernadskogo, 78
k.gaydamaka@gmail.com

Abstract: This article describes the document “INCOSE Guide for writing requirements” as a product, and the project for its translation as a focused organized activity. The article should make for the reader clear what exactly he or she can expect from the document and its translation, how to use the document in practice and how to organise projects for the translation of such documents.

Keywords: systems engineering, project management, requirements, requirements engineering, business analysis.

1 Introduction

Good textual requirements are critical for creation of successful systems.

There are many books, guidelines, manuals for writing such requirements. Unlike the INCOSE Guide for writing requirements [4], they rarely reflect the principles and processes of system engineering (for example, logical levels of a system or verification and validation of objects arising during development).

The authors hope that this article will help its readers decide if they need the Guide, and if necessary, how best to use it in their enterprise. In addition, the article should help its readers in organising translations of such documents.

To achieve these goals, the article describes the translation of the "INCOSE Guide for writing requirements" into Russian as a product [1] - describes the features of its content, presents a scenario for its use in the enterprise. The article also presents a project for the translation of this document: the principles of team building, division and automation of labor, publicity principles, rules and guidelines for translation, editing and layout. Last sections of the article contains project analysis results and respective conclusions.

The authors of the article performed following roles in the translation project: initiator and project manager, chief translation editor, translation editor, translation user.

2 Guide for Writing Requirements and its translation

The English-language Guide for Writing Requirements [4] was created by the efforts of the “Requirements” working group of the International Council for Systems Engineering (INCOSE) and was first published in July 2009. Since then, the document has undergone several reprints, which took into account the suggestions and comments of various practitioners. The latest version of the Guide at the time of this writing was published in June 2017.

The Guide translation project was initiated in 2017 and completed in 2018 by the efforts of volunteers of the INCOSE Russian-speaking chapter (project team members list can be found in the "Acknowledgments" section of this article).

2.1 Purpose and target audience of both documents

The Guide is specifically about how to express textual requirements in the context of systems engineering. The aim is to draw together advice from existing standards such as ISO 29148 and the authors and reviewers into a single, comprehensive set of characteristics, rules, and attributes. [4]

The Guide is not about the capture, elicitation or discovery of requirements, nor is it about requirements analysis. Nor is it about requirements analysis and the development of models or designs. However, requirements play a key role in these activities and processes. The focus of the Guide is:

- How to express requirements clearly and precisely.
- Record requirements in a form convenient for subsequent analysis and implementation. [4]

The recommendations in the Guide are independent of the specific IT system and are suitable for managing requirements throughout the entire life cycle of the system. [4]

The original English-language manual is devoted to writing textual requirements in English. When translating a document, the adaptation of properties, rules and attributes to the Russian language was not performed, however, many of them will be useful to those who work with Russian-language requirements. [1]

2.2 Uniqueness of the Guide

There are many guides, techniques and recommendations for writing requirements, so it is worth separately emphasizing the difference between them and the INCOSE Guide. The INCOSE Guide focuses on writing requirements in the context of systems engineering. [4] The "Introduction" section of the Guide describes the concepts of "Verification" and "Validation" specific to system engineering and their meanings in relation to requirements, designs and products. The Guide also takes the logical levels of the system-of-interest that arise as a result of architecture definition of the system (a unique process of system engineering), and the features of writing requirements associated with logical levels into account.

It emphasizes the need to take into account the context of the system-of-interest when writing requirements, that is, to take into account the following logical levels: the level of "Enterprise" (which, for example, shows the strategy of the enterprise), the level of business management and the level of business operations. Taking into account the environment in which the developed system is located is an integral part of the system approach.

2.3 Contents of the Guide

The Guide distinguishes between the requirement statements and their set. The Guide also contains the list of properties which well-written requirement statements and their sets shall possess. It also provides the list of rules, which guide the writing of requirements and help to provide the properties of well-written requirements to them. The relationships between properties and rules are explicitly described. The Guide also contains the description of "Patterns" concept for writing requirements. It is emphasized that the listed properties, rules and attributes should be tailored to the specific working conditions at the enterprise.

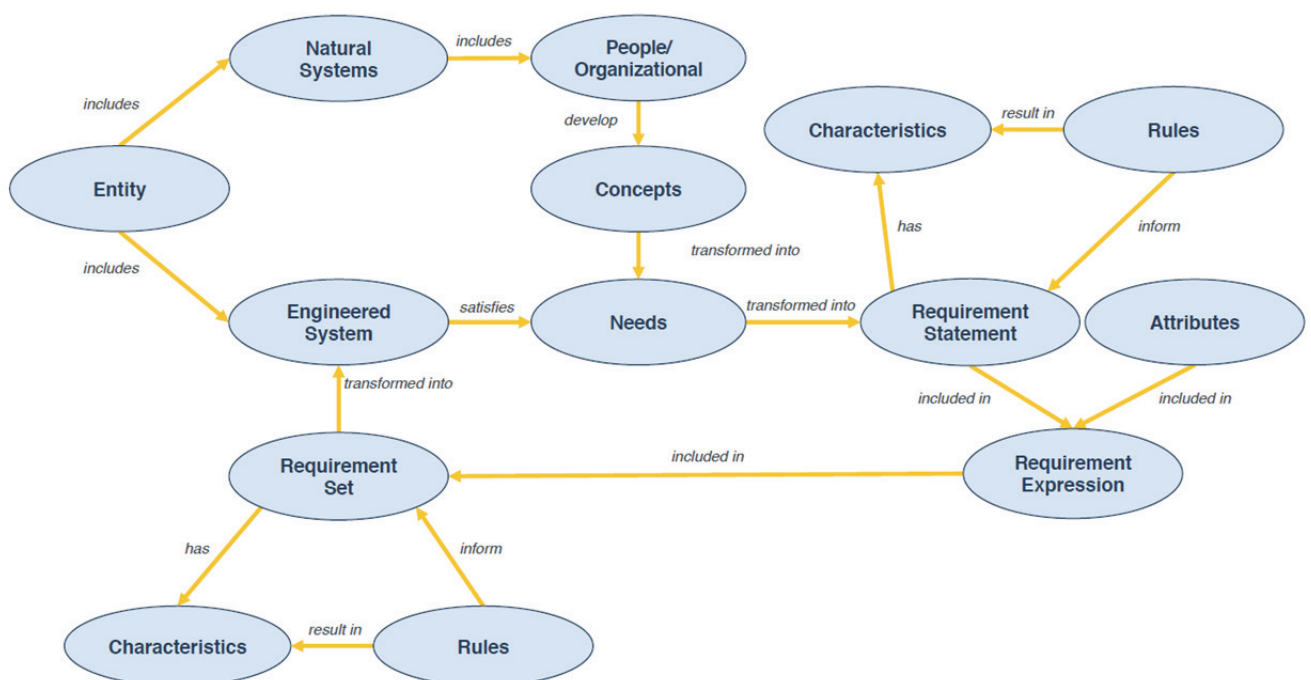


Figure 1 - A fragment of the Guide for Writing Requirements - a diagram of the relationship of entities. The contents of the Guide and the relationship links between sections are demonstrated on the figure. [4]

2.3.1 Characteristics of well-written requirement statements and sets of requirements

The Guide describes nine characteristics of well-written requirement statements and 5 characteristics of good requirement sets.

We give an example of one of the characteristics of well-written requirement statement from the Guide.

2.5 C5 - SINGULAR

Definition:
The requirement should state a single capability, characteristic, constraint, or quality factor.
Rationale:
The formal transformation is a one-to-one or a many-to-one transformation, so the resultant requirement statement must represent a single thought, aspect or expectation. The effectiveness of several processes associated with requirements, such as decomposition, derivation, allocation, tracing, and verification, depends on being able to identify singular statements. For instance, the verification information defined for a requirement can be far more precise when that requirement addresses a single capability, characteristic, constraint, or quality factor. A requirement with multiple thoughts is difficult to allocate and to trace to a parent or source.
Guidance:
Keep the requirement limited to one quality, characteristic, capability, function, or constraint. Understand how the statements fit into the traceability philosophy for the project.
Use the project standard patterns for writing requirements.
Although a single requirement should consist of a single function, quality or constraint, it may have multiple conditions under which the requirement is to be met.
Avoid the use of the word “and” when it ties together multiple thoughts (phrases in the sentence), each of which may be allocated and verified differently. The presence of the conjunction “and” in a requirement statement should always prompt the writer to consider whether or not the statement is singular.
Rules that help establish this characteristic:
R10 - /Accuracy/NoOpenEnded
R19 - /Singularity/SingleSentence
R20 - /Singularity/AvoidCombinators
R22 - /Singularity/AvoidPurpose
R23 - /Singularity/AvoidParentheses
R24 - /Singularity/Enumeration
Rules that help establish this characteristic:
R25 - /Singularity/Context
R41 - /UniformLanguage/StyleGuide

Figure 2 - Fragment of the Guide for Writing Requirements - the “Singular” characteristic description for a requirement statement. [4]

Fig. 2 shows the structure of the characteristic description of a well-written requirement statement or a good requirement set. Each characteristic has a name; each is defined and has rationale and guidelines. Each characteristic description contains links to rules that help to establish this characteristic; some characteristics have references to attributes that help to establish them.

2.3.2 Rules for writing good requirement statements and creating good sets of requirements

In the Guide, one or more rules correspond to each of the described characteristics. The fulfilment of one rule helps you to establish one or more characteristics.

In total, the Guide describes 40 rules for writing good requirement statements and creating good sets of requirements.

Here is a description example of the rules.

4.4.5 R23 - /SINGULARITY/AVOIDPARENTHESES

Avoid parentheses and brackets containing subordinate text.

Elaboration:
<p>If the text of a requirement contains parentheses or brackets, it usually indicates the presence of superfluous information that can simply be removed or communicated in the rationale. Other times, brackets introduce ambiguity.</p> <p>If the information in the parentheses or brackets aids in the understanding of the intent of the requirement, then that information should be included in the requirement attribute A1 - rationale.</p> <p>Conventions for the use of parentheses or brackets may be agreed upon for specific purposes, but such conventions should be documented at the start of the requirements document.</p>
Examples:
<p><i>Unacceptable:</i> The Control_Unit shall disconnect power from the Boiler when the water temperature exceeds 85 °C (usually towards the end of the boiling cycle.) {This is unacceptable because of the parenthetical phrase as well as the ambiguous word “usually”}.</p> <p><i>Acceptable:</i> The Control_Unit shall disconnect power from the Boiler when the water temperature is greater than 85 °C.</p> <p><i>Acceptable:</i> (if agreed for specific purposes) The Control_Unit shall disconnect power from the Boiler when the water temperature is greater than 85 °C (185 °F).</p>
Exceptions and relationships:
<p>While this rule states brackets are to be avoided, R16 - /NonAmbiguit/Logical_Condition indicates that brackets may be used as part of a convention for eliminating ambiguous conditions such as logical expressions.</p>
Characteristics that are established by this rule:
C5 - Singular

Figure 3 - A fragment of the INCOSE Guide for Writing Requirements - the description of the AvoidParentheses rule for a requirement statement. [1]

Fig. 3 shows the structure of a rule description. Each rule contains elaboration, examples of requirements written according to the rule. Exceptions and relationships to other rules are provided where there is a need of them. As in the case with the description of the requirement statement, the rule description always refers to a requirement characteristic that the rule establishes.

Note that the Guide also describes the rules for creating good sets of requirements. The structure of these descriptions is identical to the structure of the rules for the writing of individual requirements presented in this section.

2.3.3 Attributes of Requirement Expressions

In addition to characteristics of well-written requirement statements and good requirements sets, to rules for writing requirements, the Guide describes requirement attributes as a part of requirement expression. By requirement expression, we mean a record of a requirement statement in an IT system. A requirement expression consists of a statement and some requirement attributes. The Guide provides a set of attributes that can accompany a requirement statement. Attributes are explicitly described. Some attributes have references to requirement characteristics they help to establish.

Attributes are divided into four groups: “to help define the requirement and its intent”, “associated with the system of interest and requirement verification”, “to help maintain the requirements”, “to show applicability and allow reuse”. Because of the names of the groups, it is clear that the attributes are grouped by their purpose.

In total, the Guide lists 44 attributes that can be a part of a requirement statement.

5.1.5 A5 - TRACE TO SOURCE*

Each requirement must be able to be traced to its source (such as document, trade study, workshop, or interview). This is different from tracing to a parent requirement because it identifies where the requirement came from and/or how it was arrived at (rather which requirement is its parent). The source could be a stakeholder need or some form of model. Examples of sources include system concepts, user stories, use cases, regulations, standards, derived from a trade study, interviews with a stakeholder, minutes of stakeholder workshop, or Engineering Change Proposal (ECP). Sources could also be a functional area within an enterprise or business unit (marketing, safety, compliance, quality, engineering, manufacturing, etc.). Maintaining this trace is key to being able to show compliance to customer requirements, regulations, and standards. Trace to Source, along with Rationale and Trace to Parent Requirements, helps to ensure the requirement characteristic of “C1 - necessary” is met.

Figure 4 - A fragment of the INCOSE Guide for Writing Requirements - the description of the "Trace to Source" attribute. [4]

2.3.4 Requirement patterns

Once requirement writers are shown a good example of how to form a requirement statement based on its type, the difficulty in writing a properly formed requirement was largely overcome. These kinds of “structured and normalized” examples of types of requirements can be defined in patterns. A pattern may be structured as a sequential list of placeholders, including words, along with syntactic or semantic restrictions. These placeholders are generally called pattern slots. The requirement text is written per a requirement pattern that is appropriate to what is to be communicated. [4]

The Guide mentions the history of the use of patterns in projects describes the benefits of their use, presents the principles of their definition.

2.4 Use Case for the Guide

We will consider the following artificial scenario to give an example on how an enterprise can use the Guide. Let us assume that an enterprise develops, manufactures and supplies software-intensive hardware systems. The analysis of the enterprise’s projects revealed regular difficulties in the development and testing associated with the quality of textual requirements. Enterprise leadership initiated an enterprise development project, the purpose of which is to improve the quality of textual requirements and to prevent regularly arising difficulties in the development and testing.

The enterprise creates and maintains internal guidelines; its employees actively use these guidelines.

2.4.1 Elicitation of needs in the enterprise

For those who are familiar with the GOST 34 (ГОСТ 34) series of documents [6, 7], this stage is called “Formation of Requirements” (“Формирование требований”).

The first step is to determine if there is any need in the enterprise that could be met by the Guide.

In the artificial scenario described at the beginning of the section “Use Case for the Guide”, the enterprise identified regular difficulties caused by poor quality of textual requirements. Enterprise leadership decided to implement an enterprise development project.

Identified difficulties arise in the "Enterprise" system. Based on these difficulties, it is possible to formulate requirements for the "Enterprise" system. The top-level requirement (expressed, for example, in a strategic decision or in an organizational and administrative document) can be formulated as follows: “The quality of the text requirements developed at the enterprise should be such that the difficulties associated with the requirements during development and testing are minimal.”

For a strategic level, this is a good requirement. However, this requirement shall be clarified before implementation: the development project team has to define quantitative values, determine the elements of the enterprise, break down original top-level requirement and apply new detailed requirements to the elements of the enterprise, etc.

The development project team shall find someone who clearly expresses and supports the top-level need or difficulty. This person shall be capable of supporting the development project.

2.4.2 Solution trade-off analysis

In the language of GOST 34 (ГОСТ 34) standards, this stage is called the "Concept" ("Концепция") [6, 7].

The requirement to the "Enterprise" system should be implemented as part of a focused activity (for example, as part of an enterprise development project). There shall be a sponsor of this project. Sponsor is such a stakeholder representative who clearly expresses and support the need and the difficulty and is capable of supporting the project in the enterprise. Depending on the organisation, this support can be expressed in a budget allocation for the development project, in a publication of an organisational and administrative document - in the way that the organization is accustomed to highlighting and formalising any activity. The sooner such support appears, the better it is for the success of the project.

If organisational conditions allow doing this before the formal start of the project, you can analyse the needs and requirements and choose the best fitting solution - a kind of preparation for the project kick-off. Because of this work, sources of needs and requirements should be identified, needs and requirements themselves should be analysed and recorded. Project team shall propose several solutions for the set of needs and requirements. Next, the project team should perform a trade-off analysis and choose the most suitable solution.

The degree of formalization of these works and reports on them is determined by the rules adopted by the enterprise or, in their absence, by customs. In any case, a reference to a known norm or a guideline in accordance with which work is performed (for example, GOST 34 (ГОСТ 34) [6, 7]) will never hurt.

Let us suppose that the creation of a methodology for writing requirements at the enterprise was chosen as a result of the trade-off analysis. As a template for the methodology, it was decided to use the INCOSE Guide for writing requirements.

2.4.3 Formation of technical specification

If we continue to draw analogies with GOST 34 (ГОСТ 34) [6, 7], this stage corresponds to the "Technical Specification" ("Техническое задание") phase.

In order to transform the template (the Guide) into a methodology that will benefit the enterprise, it is necessary to determine what constraints and realities of the enterprise must be taken into account. That is, to identify additional requirements to the "Enterprise" system - to the part of it that is responsible for the formation of requirements for the "Product of the Enterprise" system. To do this, we should study the environment of this part. Depending on the scope of the project and the capabilities of the project team, we will have to answer various questions, for example:

- What processes are associated with the process "Development of requirements for the product of the enterprise"?
- What skills are needed now to perform the process "Development of requirements for the product of the enterprise" and will be needed in the future?
- What is the reason for the difficulties in developing and testing the product?
- What software supports the process "Development of requirements for the product of the enterprise" now and what difficulties are associated with this?
- In what organisational conditions do people who perform the process of "Development of requirements for the product of the enterprise" work and what difficulties are associated with this?

The obtained requirements for the "Enterprise" system are the basis for creating a useful methodology for writing requirements to the enterprise product. That is, the basis for adapting the Guide to the realities of the enterprise.

2.4.4 Designing Solution

This stage corresponds to the stages "Outline design" ("Эскизный проект") and "Technical design" ("Технический проект") from GOST 34 (ГОСТ 34) [6, 7].

Let us suppose that the enterprise revealed a lack of logical levels of a system: this concept is not supported in the enterprise. This is possible if an enterprise does not perform the process of "Definition of the architecture of an enterprise product" ("Architecture Definition" according to ISO 42010 [8, 9] and ISO 15288 [10, 11]). In this case, the rules and properties of writing requirements related to system logical levels (for example, the "Necessary" property) will need to be formulated differently than described in the Guide. Some attributes from the Guide (for example, "Trace to parent requirement") are likely to lose their original meaning.

Since the deployment of the process "Definition of the architecture of an enterprise product" is outside the scope of the project to create a methodology for writing requirements, the project team must adapt rules and properties that are based on the presence of system logical levels. The project team should note all the opportunities for the development of the enterprise, which cannot be satisfied within the project - in this case, the deployment of the process "Determining the architecture of an enterprise product". The project team should bring this opportunity to the attention of decision makers that can initiate development projects in the enterprise.

The previous paragraph describes design decisions that relate to the methodological part of the developed system. If there are other aspects of the project identified in the previous steps (for example, software and hardware, staff training, organisational structure), these aspects should also be addressed in the design.

Let us say the software is within the project. Then, in addition to defining the methodological rules for writing requirements, one should also determine the composition of the attributes that shall accompany every requirement statement. The INCOSE Guide to writing requirements can also serve as basis for this task (see the section "Attributes of Requirement Expressions").

2.4.5 Deployment and Testing Preparation

In GOST 34 (ГОСТ 34), this stage is called “Working documentation” (“Рабочая документация”) [6, 7].

Based on the adopted design decisions and recorded requirements, the program and methods for testing should be determined. Tests shall identify whether the developed solution really benefits the enterprise. It is better to involve in the tests those stakeholder representatives that were responsible for requirements expression. It is also highly recommended to involve end users of the developed solution in the tests - those ones, who will use the developed method and related artefacts (i.e. software, documents etc.)

Project team shall take opinion of testers seriously; testers shall have direct impact on the solution. It is also important to maintain the balance of stakeholder interests. Project team must prevent a situation in which the interests of one side are placed above the interests of the other. For example, the commercial or image interests of a project to develop a methodology should not be more important than the professional interests of end users.

2.4.6 Deployment

This stage corresponds to the stage of "Deployment" ("Ввод в действие") in GOST 34 (ГОСТ 34) [6, 7].

After successful testing and elimination of critical deficiencies, the developed guidelines for writing requirements should be formally released. Here, as at other stages, the specifics of the enterprise should be taken into account: in some, it is necessary to publish a special organisational and administrative document, while in others a verbal agreement may be sufficient.

Even if the project did not include employee training, it is recommended to conduct a short briefing on the new methodology. This, as well as testing with involvement of end-users, will help get feedback and answer questions from stakeholders.

2.4.7 Maintenance

This stage corresponds to the “Maintenance” stage ("Сопровождение") in GOST 34 (ГОСТ 34) [6, 7].

Even if the project for the development of guidelines for writing requirements does not include the support phase, it is necessary to take into account that normative and methodological documents must be maintained and have their own life cycle.

Parts of the system of normative and methodological documents developed and put into operation should be transferred for support to the relevant stakeholders. At the same time, these stakeholders shall receive information necessary for maintenance of the developed documents. Here, as at other stages, it is necessary to take into account the specifics and cultural customs in the enterprise.

2.4.8 Decommissioning

An enterprise may no longer need a system that includes guidelines for writing requirements. When developing a system, criteria and methods for identifying such situations should be defined, as well as decommission methods. When this information is gathered, the project team has to communicate it to the relevant stakeholders.

3 The Guide for Writing Requirements translation project

The Russian-speaking chapter of INCOSE (INCOSE RUS) initiated the Guide translation project in April and de-facto started in September 2017. Scope of the project included following goals:

- Release such a Guide in Russian, which will be helpful to the professional society.
- Introduce volunteer translators and professional reviewers to each other.
- Attract new members to the chapter.
- Give impetus to similar projects.
- Introduce professional translators and volunteers to each other.

According to the initial plan, the project was supposed to be completed 8 months after launch.

The project involved more than 15 people living and working in different cities of the world: in Moscow, Samara, St. Petersburg, Nuremberg, Nizhny Novgorod. Colleagues from INCOSE, living and working in the USA, France, and South Africa, supported the project.

An important principle of the project was transparency. To support this principle, the team created a website that clearly described project goals, objectives, roles, working methods and resources. The project manager divided the labour in advance: there was a clear and transparent role model in the project.

The translation project was volunteer: in fact, the project did not have a budget in terms of money. Project manager had to work hard to answer the question: “Why even participate in such a project?” The motivation aspect of team members will be highlighted in the appropriate subsection.

The translation and editing work was subject to clear guidelines for technical texts in Russian. Automation tools supported project activities (such as translation, editorial work, validation of the document by future users).

Project team had to review Soviet and Russian books on engineering to define layout rules for the layout designer. Translation, editing and layout guidelines used in the project are described in the corresponding subsection.

The project lasted 13 months and ended with the transfer of the translated document to layout in August 2018. In November 2018, the document was published. Not all of the intended goals have been achieved. To avoid such plan shifts and problems with achieving goals in future similar projects, project team conducted lessons learned session. The results of this session are described in the corresponding subsection.

3.1 Division of labour: structure of work, roles and responsibilities

The project had the following structure of work:

- Project management (planning, monitoring the implementation of the plan and updating the plan, team organization, selection of work technologies, definition of working methods).
 - Translation (transformation of an English text into a Russian text without loss of meaning and so that the result of the work is perceived by a native speaker as it would have been originally written in Russian).
 - Editing (integration of segments created in scope of “translation” work; check of the uniformity of the use of terms; additional check for semantic compliance with the source text).
 - Validation of the translated Guide by users of the document (check if the translation covers end users' needs and interests);
 - Document layout (layout of the translated document in accordance with the recommendations on the layout of Russian-language technical texts).
 - Publication of a document (providing access to a digital layout for INCOSE members and the general public through the INCOSE website; distribution of information about the release through INCOSE RUS website and INCOSE RUS social media).
 - Sponsorship (providing free INCOSE membership to volunteers who took part in the project).
- Team members distinguished between the following project roles:
- Translator (performs the work of “translation”, must have basic knowledge and skills in the subject area, the skills of an operative search for information and understanding of new concepts and their interconnections; shall use guidelines of translation adopted in the project).
 - Editor (performs the work of "editing").
 - Chief Editor (performs the work of “editing”, defines rules for translation and editorial work, divides labour between editors)
 - User of the document (validates the translated Guide).
 - Project manager (performs the work of “project management”).
 - Project sponsor (performs the work of “sponsorship”).
 - Layout designer of the document (performs the work of “document layout”).

3.2 Motivation of the project participants

All team members were volunteers. The main motivator for participation in the project was the satisfaction of professional interests during the work and because of the application of the created product.

To plan the benefits that project participants can get from interacting with each other, N2-table was used. At the intersections of roles, the benefits were indicated that the performer of the role indicated in the row would get from working together with the performer of the role indicated in the column. Project manager discussed these benefits during interviews with potential project participants.

Since the work was voluntary, it was especially important for the project to maintain a friendly atmosphere in the team. For this, special attention was paid to working with risks: possible difficulties were discussed in advance; team members determined together the measures to eliminate them or reduce their effect.

Authors of this article would like to emphasize the support of INCOSE EMEA and its chair Jean-Claude Roussel. Mr. Roussel, also being the Airbus CTO, agreed to sign letters of thanks to each of the project volunteers and to encourage them with an annual INCOSE membership.

3.3 Project Information Support and Transparency

The following information resources were used to provide information support and ensure project transparency:

- Project web site.
- INCOSE RUS social networks (Facebook page, telegram channel).
- Youtube channel INCOSE RUS.

Since the project was completely volunteer, it was especially important from the very beginning to describe the principles of work, the motivation of the participants and the promised benefits. One of the main resources of the project was the reputation of its participants: in case of success, all of the project members could place to their accounts the translation of a document that is important in an international community. The community, in turn, had to show that this project exists and tell about it as clearly as possible.

The three information resources mentioned in this section helped form a team of volunteers.

Personal network of the project participants played a significant role in the success: without it, the team would not have had many valuable personnel.

3.4 Guidelines for translation, editing and layout

Translation and editing were subject to clear guidelines [2]. The main requirement for the translated guide was its adequacy to the source text. An adequate translated text is such a text that a native speaker of a translation language perceives as a native speaker of an original language perceives an original text. Important attributes of an adequate translated text are its accuracy and completeness.

The remaining rules of the guidelines were divided into two groups:

1. compliance with the norms and rules of the Russian language;
 - 1.1. indistinguishability of paronyms;
 - 1.2. pleonasm;
 - 1.3. tautology;
 - 1.4. selection of plural / singular forms;
 - 1.5. matching words in a sentence;
 - 1.6. the creation of verbal nouns;
 - 1.7. stringing of the same forms;
2. compliance with the standards of Russian-language technical literature;
 - 2.1. impersonal appeal to the reader;
 - 2.2. high quality language;
 - 2.3. decreased imagery;
 - 2.4. construction of phrases of medium complexity;
 - 2.5. a tendency towards clarity and concreteness;
 - 2.6. a tendency to displace transliterations.

A volunteer of the INCOSE Central performed the layout designer role. Project team created a special document with layout guidelines of the Russian-language technical literature [3]. To identify the requirements for layout, we analysed Soviet and Russian books on engineering published until 1991, inclusive. The following is a list of some of them:

- Системотехника. Введение в проектирование больших систем. Г. Х. Гуд, Р. Э. Макол. М. — 1962 «Советское радио»;
- Формирование технических объектов на основе системного анализа. В. Е. Руднев и др. М. 1991 «Машиностроение»;
- Управление и творчество при проектировании систем. А. Uilson, Marthann Elizabeth Wilson Советское радио, 1976.

The following are some of the layout requirements that were submitted to the layout designer:

1. the paragraph begins with a red line of 2 em.
2. list items are marked with a long dash (alt code 0151).
3. the text of the list item is separated from the left border of the text block by a distance of 2 em (see also requirement 1);
4. list items, except the last one in the list, must end with the symbol “;”;
5. the last element of the list must end with the symbol “.”;
6. when typing quotes, the symbols “«” and “»” shall be used (alt codes 0171 and 0187 for opening and closing quotes, respectively).

These and other layout rules can be found in the document at the link: <http://tiny.cc/x5gecz>.

3.5 Translation, editing and project management tools

To support the works of “translation”, “editing”, “validation of the translation by users of a document”, the Google Translator Toolkit computer-aided translation (CAT) system was used.

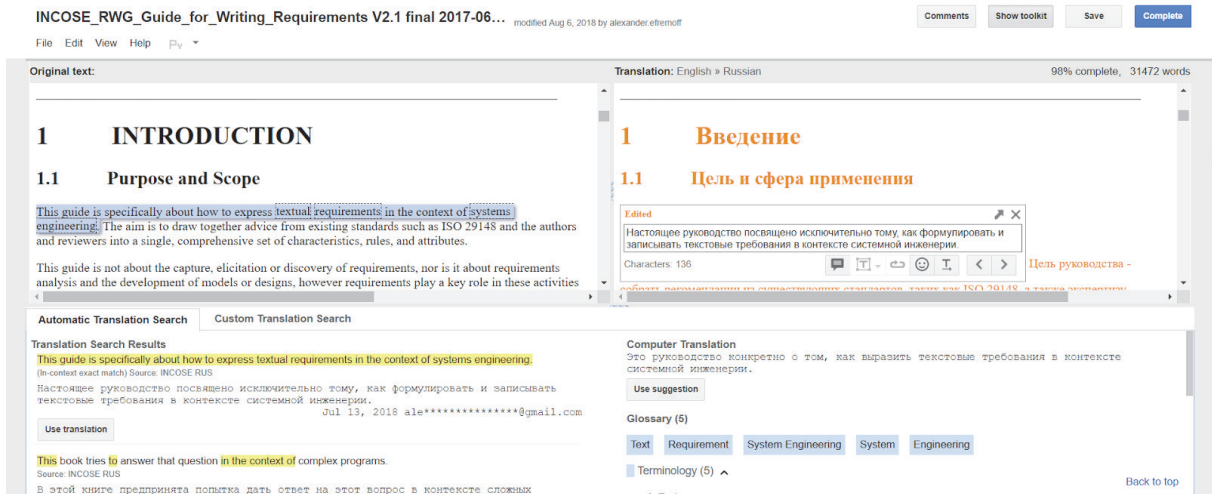


Figure 5 - Google Translator Toolkit interface. On the left side of the screen is the text in the original language. On the right side of the screen is the text in the target language. In the lower part auxiliary means “translation memory” and “glossary” are situated. The active segment - the segment under consideration by the translator - is highlighted.

Like other systems of this class, the Google Translator Toolkit supports working with translation memory and with a glossary. These tools were used as part of the project. Translation memory (pairs of segments of text in the original language and the language of translation) can be used in the following similar projects.

An important requirement for the computer aided translation system in the project was simultaneous access to the text for several project members. Each of the participants should be able to translate segments of the text, leave comments on the segment, and edit the translated segments of the text. At the same time, the system should not require additional actions on the part of users: simultaneous work should not be different from normal work with a document. The Google Translator Toolkit has met these requirements.

Another important requirement for the computer-aided translation system in the project was support of multiple platforms without additional software installation. This requirement was also satisfied: users worked with the Google Translator Toolkit through a browser from various operating systems (Windows, MacOS, Android).

The risks associated with working with the Google Translator Toolkit include slow support. Project team had issues with commenting segments of the text during several days. This affected the editing and verification of the translation by users of the document. However, these shortcomings did not have a significant impact on the project.

At the time of writing this article, Google decided to close the Translator Toolkit. Still, there are many computer-aided translation systems on the market, including web-based and free for use, which would cover the project needs.

To manage tasks, the Trello tool was used. The requirements for the task management tool in the project were simultaneous access of several users, platform independence, and ease of access. Trello met these requirements: users also worked with it through a browser and through mobile phones from various operating systems.

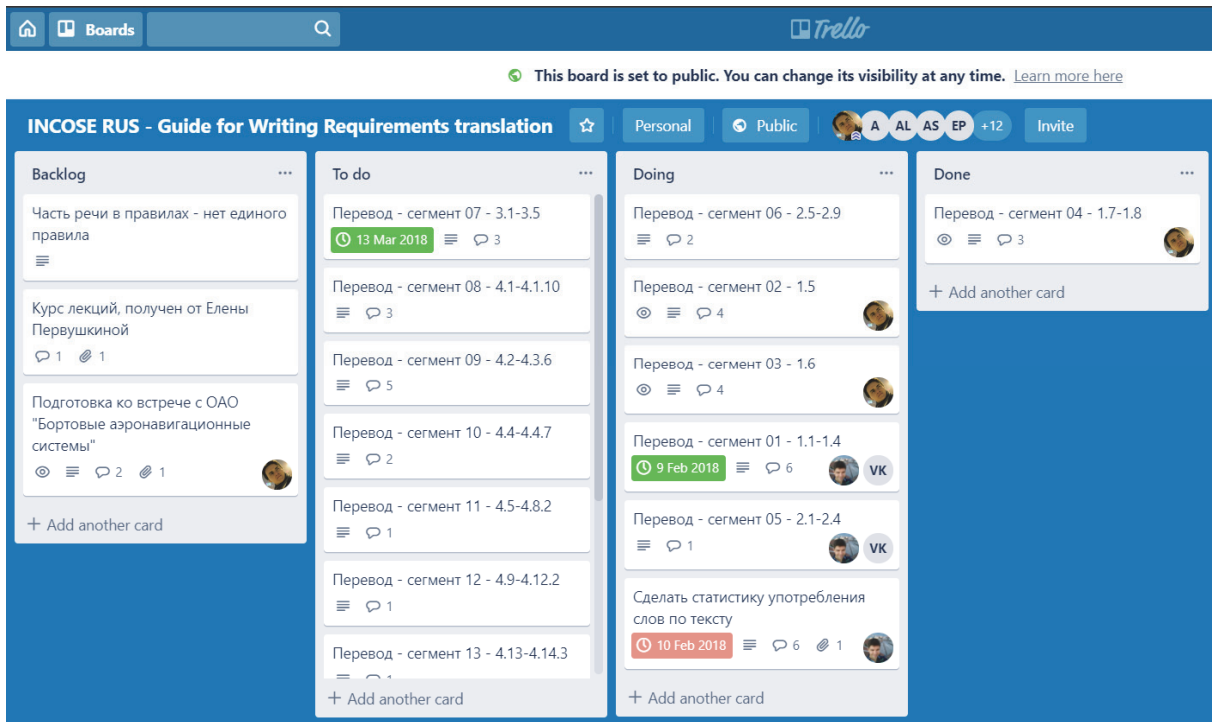


Figure 6 - Trello interface. Each task is presented in the form of a card. Cards are arranged in accordance with the state of the task in the decks: Backlog (for unallocated tasks that may have to be completed), To do (tasks that have been decided to be completed), Doing (tasks performed with assigned performers) and Done (completed tasks).

To manage comments during the translation and editing stages, in addition to the comments of the Google Translator Toolkit, the Airtable database was used. A table of translation rules has been created in the database (see the section of the article “Guidelines for Translation, Editing and Layout”). Use case for this tool chain looked like this: the editor-in-chief created a comment on the translation into the Google Translator Toolkit, created an entry in the corresponding Airtable table, and attached one or more violated rules to it. Then he linked the Airtable entry to the Google Translator Toolkit comment with a hyperlink. So translators and editors could correct the remark, guided by certain rules.

ID	Тип нормы	Описание нормы	Комментарии пользователей	Описание...	Где упомянуто
1	Адекватность перевода	Адекватность является первостепенным требованием к переводу. Неадек...	требуемыми низкими к процессу внедрения		Раздел 5.2.2.
2	Соответствие нормам и правила русского литературного языка	Критерий, позволяющий отличить подстрочник от так называемого "лите...	требуемыми к процессу внедрения некоторо...		Раздел 5.2.5.
3	Соответствие нормам русскоязычной технической литературы	Каждая область деятельности, в том числе и техническая литература, име...			Раздел 5.2.6.
4	Неразличение паронимов	Паронимы (сходные по смыслу слова) часто употребляются неправильно. ...			Приложение А
5	Плеоназм	Плеоназм - речевое излишество, вкрапление слов, ненужных со смыслов...	Процесс анализа бизнеса инициирует руководи...		Приложение А
6	Тавтология	Тавтологией называется определенное, повторяющееся в иной форме сказан...	правила для написания требований прежде вс		Приложение А
7	Выбор форм множественного/единственного числа	Часто возникают проблемы употребления единственного или множестве...			Приложение А
8	Согласование слов в предложении	При переводе часто возникают ошибки согласования слов в предложени...			Приложение А
9	Создание отлагательных существительных	Следует с осторожностью создавать отлагательные существительные. Мног...	нет ограничения понятий для выражения		Приложение А
10	Нанизывание одинаковых форм	Следует избегать нанизывания одинаковых падежных форм, например с п...			Приложение А
11	Безличное обращение к читателю	Самое значительное отличие англоязычных технических текстов от русск...			Приложение Б: нормы
12	Высокое качество языка		для новичков полезными окажутся ориентиры,		Приложение Б: нормы
13	Снижение образности		pre-defined pattern требуемых концепций да		Приложение Б: нормы
14	Построение фраз средней сложности		На данном уровне эти требования могут быть		Приложение Б: нормы
15	Тенденция к четкости изложения и конкретности		указанными ранее		Приложение Б: нормы

Figure 7 - Airtable interface, table with translation and editing rules.

At later stages of the project, project team decided to stop maintaining comments in Airtable due to the reorganisation of the project (the editor-in-chief took on all the editing functions; the performer of this role did not need additional indications of non-compliance to translation and editing rules).

Google Spreadsheet was used to plan the project and track progress.

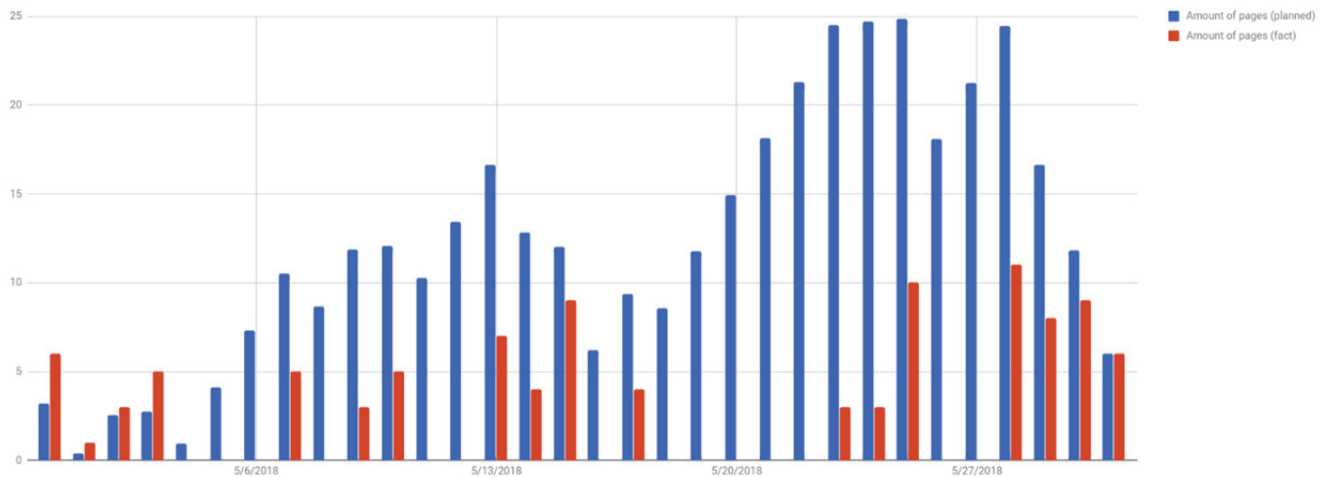


Figure 8 - Graph of tracking the status of the work "Editing", based on the data in the Google Spreadsheet. Blue indicates the planned values (the number of pages that need to be edited per day), red - the actual amount of edited pages at a day. Every day, the planned value was automatically updated based on the work performed on the previous day.

3.6 Project Summary and Lessons Learned

The project achieved two main goals: the document useful for professional community was published in Russian, new members were attracted to the community. However, the project faced several difficulties.

The project was completed 5 months later than planned (lasted 13 months instead of the planned eight).

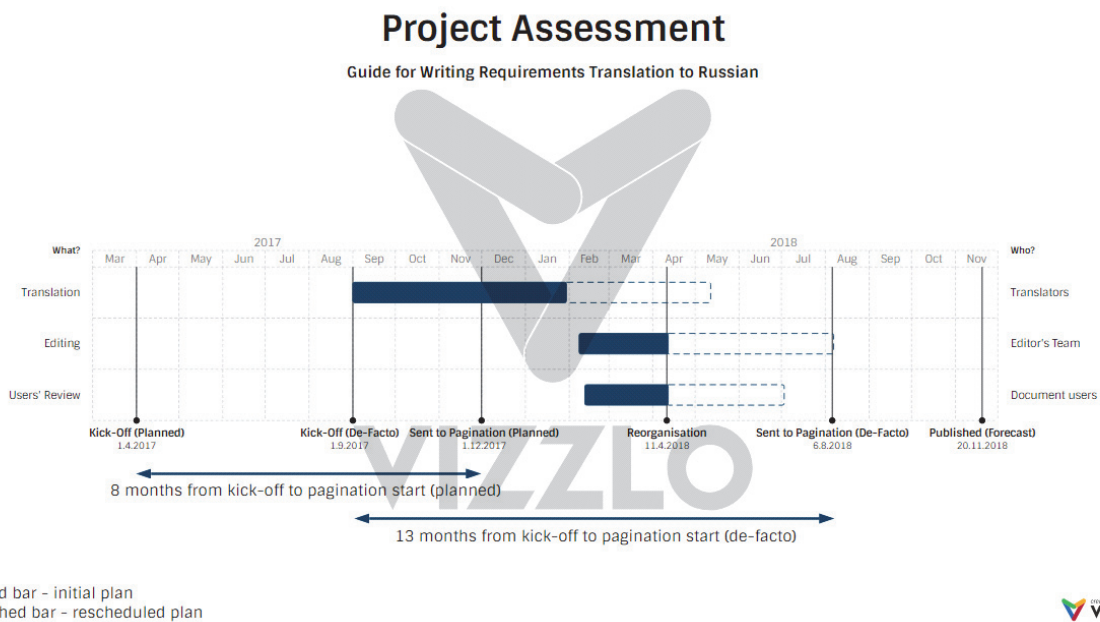


Figure 9 - Schedule plan of the project. The solid shaded rectangles indicate the initial planned dates; the dotted line indicates the actual dates.

Unfortunately no further similar project followed the original one (so impetus was not given to such projects at INCOSE RUS), professional translators didn't meet volunteer translators, volunteer translators didn't meet professional reviewers.

Project manager had to reorganise the team to complete the project.

3.6.1 Planning and Team Reorganisation

The initial plan of the project involved the phased translation, editing and validation of the translation by end users. According to initial plan, the work should have been performed in the order shown in the Fig. 10 (below).

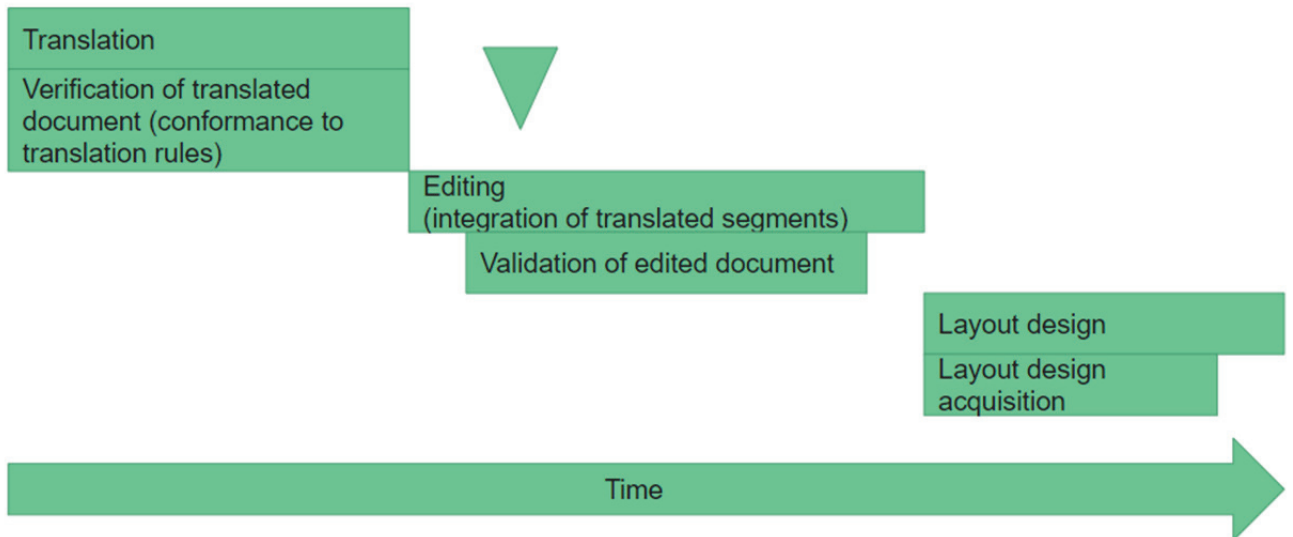


Figure 10 - Scheme of the project work dependencies, before the restructuring of the team (planned). The triangle shows the location of the decision to restructure.

That is, translators had to do the work in parallel with each other. After the text was translated completely it passed to the next phase - editing. Validation of the translated and edited text was performed in parallel with the editing work.

After the first stages of editing and validation, it became clear that the translated document does not meet the rules established in the project. To remedy the situation, it was decided to reorganise the team: the performer of the role of editor-in-chief and the editors took on the role of translators.

This problem could have been avoided by organising the work differently. If editing and translation would have carried out in parallel, translators would have immediately received feedback and could have corrected deficiencies in the text sooner. In fact, after the reorganisation, the work plan began to look just like this: translation and editing were carried out in parallel, but with different and more limited resources.

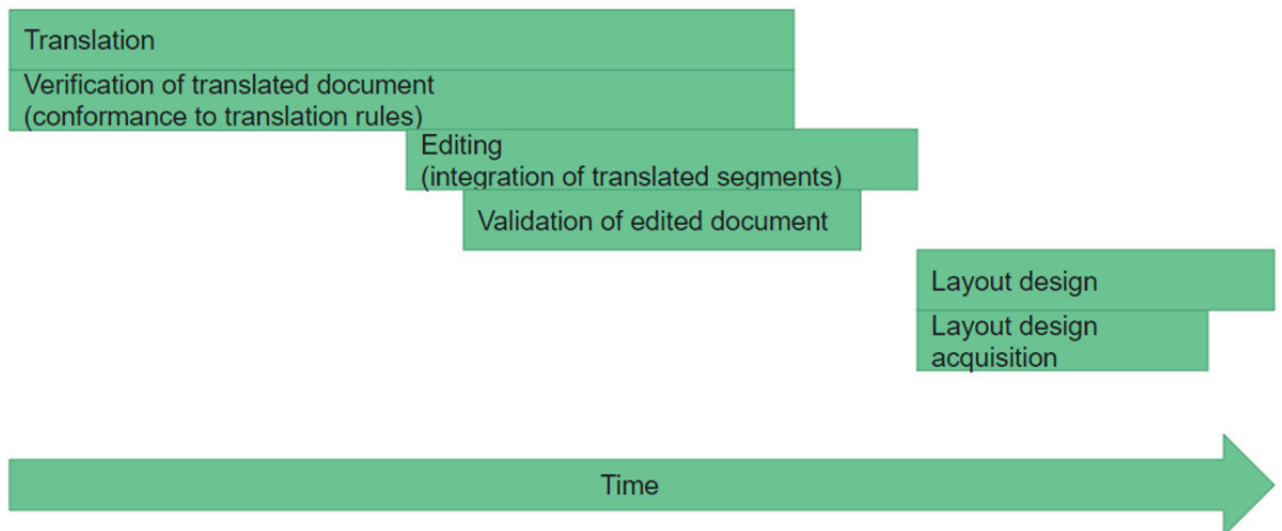


Figure 10 - Scheme of the project work dependencies, after restructuring the team (actual).

The initial work plan is suitable for well-coordinated teams that are well aware of the rules of editing. For a team consisting of people who have never worked with each other and with the rules in the project, it is necessary to provide feedback and communication at the earliest possible stages.

An additional difficulty for translators was the low readiness of the glossary. However, errors in translation of the terms were insignificant in comparison with the violation of the translation rules adopted in the project.

Some numbers

412

(Amount of issues reported by users of the document and processed during the project by the team of editors)

2100

(Amount of document corrections: difference between translation output and design layout input)

Figure 12 - Some numbers of the project: the number of issues reported by users of the document (top-level comments, many of which had considerable discussions) and the total number of corrections made in the document that was received from translators. Total volume of the document: more than 31,000 words.

3.6.2 Key lessons learned

After lessons learned session and examining the project results, the project team identified the following conclusions:

- for a team consisting of people who have not previously worked with each other: at the beginning of the project, it is necessary to introduce people to each other;
- if project team members are not familiar with methods and technologies adopted in the project: spend more resources on training, plan parallel work and assure feedback between the functions in the team;
- greater control of agreements with sponsors by the project manager;
- it is necessary to pre-communicate possible risks with team members and jointly determine measures to eliminate the consequences;
- as a special case of working with risks - the whole project team shall be ready for reorganisation in critical situations while maintaining companionship between people;
- glossary shall be completed as early as possible;
- the labor of the project participants must be paid;
- translation project team shall have expertise in the subject area.

4 Opportunities for future projects

Firstly, the published translation is not adapted to the specifics of the Russian language. From the very beginning, this task turned out to be outside the scope of the translation project, as it required a more complex organization, the involvement of linguists and practicing requirements engineers.

Secondly, the translated and published text can be improved in terms of ease of perception. This is a relatively simple job, which should involve a literary editor.

Based on the properties of the good requirements described in the Guide, at least one commercially available software has been created that automatically evaluates text requirements (see <https://www.reusecompany.com/>). This commercial solution does not yet support working with Russian-language text requirements. Adapting the Guide to the Russian language and automating the verification of text requirements is an opportunity for a separate project.

Project team created translation memory during the project (translation memory contains pairs of sentences in the original language and the target language), a glossary. This data can be used in the following translation projects.

Developing a glossary is a separate, time-consuming job. The existence of such a glossary, which would indicate terms specific to system engineering in English, their translations in Russian with links to various sources (the context of the use of these words) would be useful not only in the scope of a translation project, but also in the engineering community in general. As a basis for the glossary, the project used the Book "Словарь-справочник" [5] by Professor Viktor Konstantinovich Batovrin. Creating such an interactive document is an opportunity for a project.

Among the published INCOSE products, many English-language documents will be useful to the Russian-speaking engineering community. Translation of each of them is a separate possible project.

5 Acknowledgements

The authors of the article are grateful to all the participants in the project and the people who supported it.

The authors of the article express special gratitude to the following INCOSE colleagues, without whose support it would have been much more difficult to complete the work: Mike Celentano, Jorg Lalk, Rachel LeBlanc, René Oosthuizen, Kerry Carey, Jean-Claude Roussel, Amaury Soubeyran.

Project team members:

Project Manager, Editor-in-Chief:

- Alexander Efremov, Piterion GmbH, Nuremberg, Germany.

Editors:

- Kirill Gaydamaka, JSC "International Aeronavigation Systems Concern" (IANS), Moscow, Russia;

- Valentina Kurochkina, Moscow, Russia.

Translators:

- Valery Evseev, NRNU MEPhI, Moscow, Russia;

- Anton Korolev, NRNU MEPhI, Moscow, Russia;

- Elena Pervushkina, JSC Concern Titan-2, St. Petersburg, Russia;

- Tatyana Poletaeva, NRU Higher School of Economics, Nizhny Novgorod, Russia;

- Anastasia Soboleva, NRNU MEPhI, Moscow, Russia;

- Oksana Tishina, NRNU MEPhI, Moscow, Russia;

- Tatyana Fomina, NRNU MEPhI, Moscow, Russia.

Translated Guide end-users (reviewers):

- Vladimir Volkov, LLC "Форсайт Про", Moscow, Russia;

- Vsevolod Ivanov, LLC "GRADUM", Moscow, Russia;

- Alexander Luchkov, Concern "International Air Navigation Systems" JSC, Moscow, Russia;

- Dmitry Pinaev, GC Modern Management Technologies (ГК «Современные технологии управления»), Samara, Russia.

Bibliography

- [1] Russian Translation of the INCOSE Guide for Writing Requirements. <https://connect.incose.org/Pages/Product-Details.aspx?ProductCode=RequirementsRussian>
- [2] Requirements for the provision of text by freelance translators. Work instruction РИ7.4.02-01 / 03. Developer E. Rembovskaya. "БТД Неотек" LLC (Neotech). Moscow, 2004. -- 56 p.
- [3] Recommendations for the layout of Russian-language technical literature; Developer A. Efremov. INCOSE RUS. <http://tiny.cc/x5gecz>
- [4] Guide for Writing Requirements, INCOSE-TP-2010-006-02.1, June 2-17. <https://connect.incose.org/Pages/Product-Details.aspx?ProductCode=TechGuideWR2019Soft>
- [5] Системная и программная инженерия. Словарь-справочник. (Systems and Software Engineering. Dictionary and reference book) Виктор Константинович Батоврин (V. K. Batovrin). Litres, 2017
- [6] GOST 34.601-90. Automated systems. Automated systems. Stages of creation. (ГОСТ 34.601-90. Автоматизированные системы. Стадии создания)
- [7] RD 50-34.126-92 (РД 50-34.126-92) Recommendations. Information technology. Guidelines for creating automated systems.
- [8] ISO/IEC/IEEE 42010 Systems and software engineering — Architecture description
- [9] GOST R 57100-2016 (ГОСТ Р 57100-2016) System and software engineering. Architecture Description
- [10] ISO/IEC/IEEE 15288:2015 Systems and software engineering — System Life Cycle Processes
- [11] GOST R 57193-2016 (ГОСТ Р 57193-2016) System and Software Engineering. System life cycle processes