



TEACHING ENGLISH WRITING FOR SPECIFIC PURPOSES

N. Dychka

*National Technical University of Ukraine “Kyiv Polytechnic Institute”
ndichka@yandex.ru*

Abstract

In this paper we study the problem of teaching writing in English for Specific Purposes. The research deals with how to teach the students of computer sciences to create such an important document for them as Software Requirements Specification (SRS) in an effective way. We characterize the language quality of a SRS, its composition, lexical, grammatical and syntactical features of a SRS. We identify the main theoretical approaches to teaching writing a SRS: process approach and student centered approach. The effectiveness of teaching writing the SRS to the students of computer specialties can be achieved through the creation of a rational system of exercises using elements of e-learning as it is natural environment for them. We propose the system of exercises “SRS Development” (on the platform Moodle) which is divided into groups of exercises aimed at mastering certain skills and abilities in writing.

Key words: English, teaching writing, elements of e-learning, students

Introduction

We are witnessing a rapid development of a software industry in Ukraine. There are five dominant leaders in the development of a software: the U.S.A., China, Russia, India, Ukraine. That is why arises the question of qualified English language training for the students of computer specialties, which is aimed to create skillful professionals, competitive in the world of software market developers.

Thus, English language becomes an essential component of the Ukrainian higher education system of teaching students of computer specialties.

Any development in the field of technical industry is a strictly regulated process that consists of certain steps and sequences made by a designer. Creating a new product (device, software, information technology, etc.) begins with the development of a special document called Software Requirements Specification (SRS). Students which study computer sciences in Ukrainian Universities have the task to present their course project (2nd, 3rd year of studying), bachelor project (4th year of studying), magister project (5th, 6th year of studying). These projects start from writing a SRS. And after the university graduation, a specialist, a professional is involved in the development of various specifications as a software developer or a customer.

The productive activities of a specialist in the field of information technology (IT) are often connected with foreign partners – firms, companies, offices. The market of IT services is internationalized, so the skills and abilities to write a SRS in English are essential attributes of the expert’s successful performance in the field of production.

Therefore, the development of teaching methods of writing for specific purposes (in particular, writing a SRS) is an important task in the training of highly qualified specialists in computer sciences.

Overview

Pierre Bourque and Robert Dupuis (2004), Don Widrig (2003), Roger Pressman, (2010), Ellen Gottesdiener (2009) give such a definition of a SRS: Software requirements Specification (SRS) – a [requirements specification](#) for a [software system](#) – is a complete description of the behavior of a system to be developed and may include a set of [use cases](#) that describe interactions the users will have with the software. In addition it also contains [non-functional requirements](#). Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

SRSs are ideal projects for technical writers to be involved with because they lay out the foundation for the development of a new product and for the types of user documentation and media that will be required later in the project development life cycle (Leffingwell, 2003).

The SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an e commerce web site, and so on) must provide, as well as states any required constraints by which the system must abide. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the “parent” document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it (Pressman, 2010).

The language quality characteristics of the SRS:

Complete: A SRS defines precisely all the go-live situations that will be encountered and the system’s capability to successfully address them.

Consistent: A SRS capability functions and performance levels are compatible, and the required quality features (security, reliability, etc.) do not negate those capability functions.

Accurate: A SRS precisely defines the system’s capability in a real-world environment, as well as how it interfaces and interacts with it. This aspect of requirements is a significant problem area for many SRSs.

Modifiable: The logical, hierarchical structure of the SRS should facilitate any necessary modifications (grouping related issues together and separating them from unrelated issues makes the SRS easier to modify).

Ranked: Individual requirements of a SRS are hierarchically arranged according to stability, security, perceived ease/difficulty of implementation, or other parameter that helps in the design of that and subsequent documents.

Testable: A SRS must be stated in such a manner that unambiguous assessment criteria (pass/fail or some quantitative measure) can be derived from the SRS itself.

Traceable: Each requirement in a SRS must be uniquely identified to a source (use case, government requirement, industry standard, etc.)

Unambiguous: A SRS must contain requirements statements that can be interpreted in one way only.

Valid: A valid SRS is one in which all parties and project participants can understand, analyze, accept, or approve it.

Verifiable: A verifiable SRS is consistent from one level of abstraction to another. Most attributes of a specification are subjective and a conclusive assessment of quality requires a technical review by

domain experts. Using indicators of strength and weakness provide some evidence that preferred attributes are or are not present.

A composition of a SRS consists of: SRS Title Page, List of Figures, Content, Main Body, Notes, Appendixes, Index (IEEE Std 830-1998).

Main body consists of mainly 3parts:

1. **Introduction** (Purpose, Scope, Definitions, Acronyms, Abbreviations, References), Overview),
2. **Overall Description** (Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions and Dependencies)
3. **Specific Requirements** (Interface Requirements, Functional Requirements, Non-Functional Requirements, Design Constraints, Database Requirements)

Let's consider lexical, grammatical and syntactical features of a SRS. A special vocabulary, which has a terminological nature, is used in science, art, industrial production, in various professional spheres. Terms are words and compound words that in specific contexts are given specific meanings, meanings that may deviate from the meaning the same words have in other contexts and in everyday language. The term, unlike ordinary words, not only calls but also logically and accurately defines a notion, it provides the concise description of an object or phenomenon, contains more information than any lexical unit (Brooks, 1991). The texts written in scientific-technical sub-style are saturated with the terms (Sudovtsev V. A., 1989). The students' SRSs include the terms that belong to specialty "Computer Engineering" and "Software engineering": *register bits, handler, memory controller* and so on. The main requirements for the use of terms are uniqueness and brevity. As it was noted by V. A. Sudovtsev, the term for the specialist is the equivalent to the notion.

A general vocabulary is also used in SRSs: *provides, explains, defines, reviews, describes*. It helps the author to achieve the objectivity and generality.

A specific feature of a SRS is its phraseology. It doesn't have stable phrases or clauses with a stylistic coloring. But the information is very well represented with stylistically neutral phrases: *to be of importance, the sphere of application, not greater than, not less than*, etc. Also the prepositions are used in a SRS: *for, during, in, since, till*.

It is worth to mention that the same noun in the text of a SRS can be repeated even in the sentences that stand nearby and it is not replaced by a pronoun. In SRSs are often used multi-word combinations, using them the spec writer can avoid the excessiveness of information. These are the phrase, which are formed: Noun + Noun (*debugging support, memory roll-over*). Complex phrases which are composed of a series of nouns: Noun + Noun + Noun (*keyboard controller code, hardware suspend requests*); Noun + Noun + Noun + Noun (*system management status registers*).

A SRS is usually full of acronyms (e.g. IEEE: Institute of Electrical and Electronics Engineer), abbreviations (e.g. DB: Data Base), definitions (e.g. Client: Someone that purchases the final product, professional (e.g. Admin: Administrator) and Latin (e.g. - for example, etc.-etcetera) abridgements. The nouns, which name people by some characteristic, action or attitude are often met in SRSs: *user, spec writer, programmer, head of the project*.

Such special phrases as continuances (*below, as follows, following, listed*) follow an imperative and introduce the specification of requirements at a lower level. Categories of words and phrases that indicate illustrative information within the SRS are called directives (figure, table, for example, note). A high ratio of total number of directives to total text line count appears to correlate with how precisely requirements are specified within the SRS.

Grammatical means are also important to be considered. Present Simple and Future Simple in Active and Passive Voices are the most often used tenses in a SRS:

The right side of the screen behaves the same way as described previously under Home Page.

Programming is controlled by the keyboard Clock Control Register.

The vehicle's exhaust system will power the ABC widget.

The data will be held in an Access database on the departmental server.

There are a lot of Participles (Participle 1, Participle 2), Gerunds and Infinitives in SRSs:

The signal generated by this pin is a toggle.

Unused timers should be masked by setting the corresponding timer enable bit to zero.

In designing products, it helps to imagine a few real life stories of how actual (stereotypical) people would use them.

SRSs are full of modal verbs (must, mustn't, shall, shouldn't, can, can't, have to):

It should be at least 800 pixels from the left border of the animation and 600 pixels from the top.

The rigidity of the machine must be such that its deformations under maximum load are negligible.

The use of passive constructions is common for the SRSs, especially Modal + Passive Infinitive:

The keyboard controller can be divided by 2, 4 or stopped.

These bits should be programmed to be zero.

Conditional sentence can be used within the SRS:

Contact Customer Support if you get any other error.

Each local standby timer will not start counting until two events occur.

The pronouns of the first person (I, me, my) are not acceptable. Technical writing requires impersonal way of presenting information. The SRS' sentences have an impersonal feature. The position of the subject is occupied by "it" or "there":

There is no special protection built into this system.

The descriptions of the requirements for the program can be made using the following pronouns: who, which, that. They serve for connecting the parts of the text or sentences:

It displays a webpage that enables the buyer to request the display of the buyer guidelines.

Along with lexical and grammatical means the syntactic organization of a SRS has its own peculiarities. A SRS has a fairly strict compositional form. Compositional structure of the expressions in scientific-technical style is subjected to the principle of clear separation of one idea from another. The separate statements in the text are usually numbered:

Displayed when the Shockwave animation is complete, the Home Page serves three purposes:

- 1. Allow people to learn about the service and consider whether they want to sign up.*
- 2. Allow members who have already signed up to log on.*
- 3. Allow people who want to sign up to create an account.*

As for the syntax of scientific-technical texts, they have long sentences, with an extremely extensive use of conjunctions:

Each screen in WhatTimeIsIt.com is known by a canonical name which will always appear, in this document, with an underline, so you know we're referring to a screen by name, for example, Home Page.

Sometimes one sentence covers all the terms of the agreement. The relations between the sentences reflect the real connections between the distinct concepts. This relationship is particularly clear when there are appropriate conjunctions:

Because of the high similarity between the various screens, some system of includes should be used on the server so that if the name of the service changes, or if we can't purchase the domain name we want, we'll be able to change all the screens in one place.

In simple sentences: a) the subject is followed by the predicate and secondary members of the sentence; b) the attribute is followed by the defining word; c) the circumstances are closer to the defining words; d) introductory words at the beginning of the sentence:

The graphics and layout of the screens is shown here merely to illustrate the underlying functionality.

As we can see, there is a great amount of information in one sentence. The result of this is that the syntactic structure of the SRS is complicated.

The effectiveness of teaching writing the SRS to the students in computer specialties can be achieved through the creation of a rational system of exercises. It should be done with the use of technology as it is natural environment for the students in the 21st century. The research conducted in the NTUU “KPI” in 2012 showed that the majority of English teachers (86.5%) believe teaching writing is productive using elements of e-learning. The advantages of teaching writing using elements of e-learning were identified: ease of use, time saving, personal approach to learning, interactive feedback between a teacher and a student, flexibility, efficiency, student’s mobility, accessibility, increase of motivation, the teacher acts as an assistant, compliance with modern teaching methods.

So there was created the system of exercises “SRS Development” using the platform Moodle. It is used in a blended learning format.

Blended learning (BL) is a relatively new ‘pedagogical’ approach, studied by such researches as: J. Carmen, J. Bersin, M. Vlada, I. Lam, R. Taconis, R. Graham, S. Flavin, S. Allen, D. Gonzalez. There are many definitions of BL but at the core of all the definitions BL implies a combination of face-to-face and online learning as components of the pedagogy (Graham, 2006).

Potential benefits of blended learning include pedagogical richness (shifting from a presentational format to active learning), greater access to resources and experts, greater flexibility; increased interaction.

It needs to be stressed that blended learning is not just a mixture of strategies and technologies, but a holistic didactical method that combines “the effectiveness and socialization opportunities of the classroom with the technologically enhanced active learning possibilities of the online environment, rather than ratio of delivery modalities” (Alvarez, 2005).

Researchers have found that the e-learning component of the blended course gives the student a chance to revisit lessons. Because of the ability of students to self-pace, there is a higher completion rate for students in blended learning classrooms (Flavin, 2001). This self-pacing allows the engagement of every learner in the classroom at any given time. Students also see that the learning involved becomes a process, not individual learning events. This revelation allows an increased application of the learning in the classroom (Flavin, 2001). Overall, blended learning offers more affordances than constraints.

Analysis of existing sources, both practical and theoretical, has allowed us to identify the main theoretical approaches to teaching writing a SRS using elements of e-learning (program “SRS Development”) for work in a classroom and at home: process approach, student-centered approach. Let’s consider each approach.

The process approach. There are 3 stages of the SRS’ development: pre-writing, writing SRS and post-writing stages. A spec writer on each of these steps solves certain tasks. According to our methodology on the pre-writing stage a student analyzes and structures the SRS: studies the information concerning writing SRSs, analysis the sample of the SRS, defines the topic, structures the SRS according to the standard IEEE - 830, creates the input data for the SRS and so on. On the second stage, students are actually writing their own SRSs: they analyze the stylistic features, learn vocabulary and grammatical structures typical for the SRS, write 3 parts of the SRS (Introduction, Overall description, Specific requirements). On the post-writing stage, the students perform editing and evaluation of the SRS.

The student-centered approach implies the organization of the learning process, based on individual psychological characteristics of the computer specialties students (thinking, memory, perception, attention, intelligence, interests); consideration of motivation, intellectual and other general and special

abilities of the students of computer specialties; consideration of the professional training factors in order to develop a proper system of exercises.

Each methodology is based on the didactic and methodological principals of learning a foreign language. The system of exercises “SRS Development” is based on the didactic principals of scientism, availability, regularity and consistency, clarity, awareness, interdisciplinary connections, student autonomy. The methodological principals of the course are: interactivity, feedback, communication, consideration of the native language, a dominant role of exercises, integration of studying writing with other types of speech activity, variability, control.

As it was mentioned above the system of exercises “SRS Development” consists of the three subsystems, which are divided into groups of exercises aimed at mastering certain skills and abilities in writing:

1) exercises for the development of skills and abilities to design the structure of the SRS and to process the information on the pre-writing stage;

2) exercises for the development of skills and abilities of writing on the “SRS-writing” stage – groups of exercises: for the development of the skills stylistically analyse the SRS, for the development of lexic and grammar writing skills, for the development of skills to write the main parts of the SRS;

3) exercises for the development of the SRS editing skills on the post-writing stage.

The purpose of the first subsystem of exercises is to teach students to analyze the structure of the SRS on the pre-writing stage. These exercises prepare students for writing their own SRS. This means that the purpose of these exercises is to acquaint the student with the examples of good SRSs, to teach students to structure the content of the SRS, generate input data for the SRS. The following types of exercises are used in this subsystem: acquaintance with the presentation “Software Requirements Specification”, creation of the slides for the presentation, writing topics for the SRS, translation of the topics from Ukrainian into English, making notes, writing input data for the SRS, analysis and synthesis of the information after reading the sample of the SRS, creation of the SRS’ content. Some of the exercises are checked by the computer and some by the tutor.

The aim of the second subsystem of the exercises is to teach students to compose their own SRS. The subsystem includes the exercises concerning stylistic features of the SRS, vocabulary, grammar and writing different parts of the SRS. Types of exercise: matching, choosing correct answer out of 2, 3 and 4 options, filling in the tables, differentiation, creation of terms, finding certain lexical items in the sentences, the formation of one part of the speech from another, solving a crossword, filling the gaps (conjunctions, pronouns, forms of verbs), transformation, punctuation marks, putting words and sentences in the right order.

The third subsystem of the exercises aims to teach students edit their own text and texts of other students. Types of exercises: editing of sentences, passages, texts.

Conclusion

Thus, teaching students of computer specialties to write in English using elements of e-learning integrate the traditional way of learning and e-learning in the classroom and at home to achieve the aim of teaching – the development of skills in ESP writing. Elements of e-learning in the educational process have to satisfy such requirements as professional orientation, interactive feedback, usability, individualized learning, motivation, modularity, monitoring and evaluation, the new role of the teacher, new role of the student.

The advantage of using elements of e-learning in teaching students of computer specialties are multiple and undisputable: they adapt to the needs and goals of each student, allow to choose the pace for the student, provide challenging learning, act upon different perception channels responsible for apprehension of language material, reduce the duration of checking the home task, provide effective

feedback and objective evaluation of student's progress. It is obvious that blended learning is becoming the standard method for university education these days. This situation suggests the need to develop different foreign language computer programs for university students.

Bibliography

1. Alvarez, S. (2005). Blended learning solutions. In B. Hoffman (Ed.), Encyclopedia of Educational Technology. Retrieved October 10, 2005, from <http://coe.sdsu.edu/eet/articles/blendedlearning/start.htm>
2. Bersin, Josh. (2004). "The Blended Learning Book: Best Practices, Proven Methodologies, and lessons Learned. Retrieved October 2, 2005 from http://media.wiley.com/product_data/excerpt/67/07879729/0787972967.pdf
3. Flavin S. E. Learning Advantages in a Tough Economy. Retrieved Oct. 6, 2005 from <http://www.babsoninsight.com/contentmgr/showdetails.php/id/217>
4. Gottesdiener, Ellen (2009). The Software Requirements Memory Jogger: A Desktop Guide to Help Business and Technical Teams Develop and Manage Requirements. Addison-Wesley. [ISBN 157681114X](#).
5. [IEEE Recommended Practice for Software Requirements Specifications \(IEEE Std 830-1998, Revision of IEEE Std 830-1993\)](#), <http://ieeexplore.ieee.org>.
6. Leffingwell, Dean; Don Widrig (2003). Managing Software Requirements: A Use Case Approach (2nd ed. ed.). Addison-Wesley. [ISBN 032112247X](#).
7. Pierre Bourque and Robert Dupuis, ed. (2004). [Guide to the Software Engineering Body of Knowledge, 2004 Version](#). IEEE Computer Society. pp. 2–1. [ISBN 0-7695-2330-7](#), <http://www.swebok.org>.
8. Pressman, Roger (2010). Software Engineering: A Practitioner's Approach. Boston: McGraw Hill. p. 123. [ISBN 9780073375977](#).
9. Sudovtsev V.A. (1989). Science and Technology Information and translation: Textbook, 232p., Moscow.
10. Tribble Ch.(1996). Writing, p. 174, Oxford Univ. Press.
11. Brooks A. (1991). Writing for Study Purposes : A teacher's guide to developing individual writing skills, 162 p., Cambridge Univ. Press.
12. Vlada M., Jugureanu R., Istrate O. E-learning and Educational Software// Educational Projects and Experience of Implementation in Romania. http://www.icl.evu/2009/disc/icvl/documente/pdf/met/ICVL_ModelsAnd