

An Approach to Design Thesis Papers in Software Engineering

D. V. Koznov, D. M. Nikolaeva

Saint Petersburg State University, Russia

Метод проектирования дипломных работ по программной инженерии

Д. В. Кознов, Д. М. Николаева

Санкт-Петербургский государственный университет

Keywords: software engineering, mind maps, documentation, thesis.

The paper describes a method to develop thesis papers on software engineering and control their quality. The method is a course template for final year students (undergraduate and graduate ones) to be delivered by university departments interested in a higher quality of students' texts. Primary tools employed are mind maps: students make their thesis paper plans, have them approved by their advisors and course lecturer, and create first drafts of their theses. The paper describes the course structure and techniques used. The method was used in 2011–2013 by the department of software engineering, St. Petersburg State University. We present measurements of paper quality for our department theses before (2007–2009) and after the course was introduced (2011–2013).

Ключевые слова: программная инженерия, дипломы, интеллект-карты, документация.

В рамках данной работы изложен метод проектирования дипломных текстов и контроля их качества со стороны выпускающей кафедры, применяющийся на кафедре системного программирования СПбГУ с 2011 по 2014 гг. Метод реализуется в рамках обязательного для всех выпускников кафедры курса, проходящего в последнем учебном семестре, перед защитой дипломов. На этом курсе специальный сотрудник кафедры ведёт занятия, на которых студентам предложено с помощью интеллект-карт (mind maps) выполнить планы своих текстов, обсудить эти планы со своими научными руководителями и написать первую версию текста. Успеваемость студентов на этом курсе тщательно контролируется кафедрой. В работе подробно описывается структура курса и используемые методики. Также приводятся данные по измерению качества дипломных работ нашей кафедры за три года до курса (2007–2009 гг.), а также за время работы курса (2011–2013 гг.).

1. Introduction

Universities are increasingly opening their data on the Internet, providing more and more information on their academic activities and administration. Open data help to improve universities' operations and facilitate their relationships with the society, allowing assessing their performance [1]. By publishing student qualification papers universities provide public evidence of the quality of education they offer. Readers get an idea about the range of topics, the scope and depth of student research, and many other aspects of study. One can also see the changes taking place over time. Software Engineering Department of Saint Petersburg State University, for example, has been publishing theses and course papers as well as other academic information online since 2007 [2]. But what is important here is the fact that papers must be well-written to reflect the quality of research itself.

For many students, creating a good text does not appear an easy task. Theses are academic papers and are to be written according to a number of strict rules. However, it should be noted that software engineering students do not always aim at becoming professional scientists, even though they do have to comply with academic traditions and write science-oriented theses. Europe and the rest of the world are calling for involving industry in education to prevent it from being overly academic. For example, one of the goals of «Research Networks (ITN): Support for Innovative Training Networks», which is part of the Horizon 2020 initiative, is to support industrial PhD programs in which non-academic organizations play a role equal as that of universities [3]. This raises a need for new methods to help students meet academic requirements and use them to their own advantage without obtaining excessive, unnecessary skills. Being able to create academic texts is not indispensable for software engineers to work successfully in industry. Yet, students do benefit from such experience because software industry requires text-writing skills. On the other hand, availability of high quality texts on the Internet is important to assess performance of university departments.

Consequently, it would be reasonable to provide thesis paper development support for software engineering students. Our experience proves that students are reluctant to create texts and tend to underestimate the difficulty of this task. Their supervisors focus primarily on solving the research problem without paying much attention to the quality of texts. Since it is the department that is interested in better texts, it is logical to provide some text development support (Fig. 1).

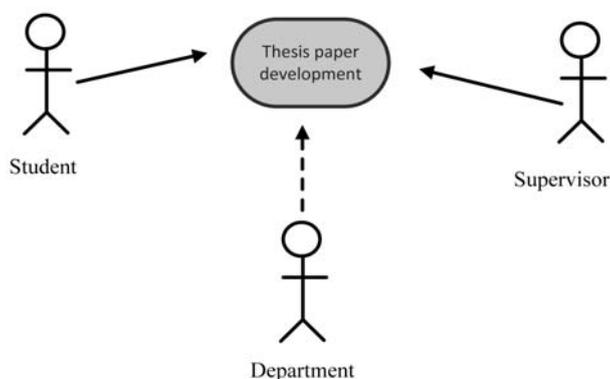


Fig. 1. Participants of thesis paper development process

In Russian universities, however, theses are submitted to departments only one or two weeks prior to defense. Our department produces about 40 graduates annually, and this short time is not enough for professors (except for a supervisor) even just to read all the papers. The problem of improving the quality of texts apparently requires a special solution.

The paper describes a method to develop student thesis papers on software engineering and control their quality. The method is a course template for final year students (undergraduate and graduate) to be delivered by the university department interested in a higher quality of students' texts. Primary tools employed are mind maps [4] and the online service of Comapping [5]. Students make their thesis plans, have them approved by their advisors and course lecturer, and create first drafts of their theses. The latter step is important not to have the plans as the only results of the work. The department closely monitors students' progress in the course. The paper describes the course structure and the techniques used. The method was used in 2011–2013 by the department of software engineering, St. Petersburg State University. We present measurements of text quality for our department theses before (2007–2009) and after (2011–2013) the course was introduced.

2. Related works

Many courses on software engineering train students to create technical documents: see [6] and [7] to review such courses with their own tools and methods. There is also a variety of system approaches to writing user documentation [8] and a lot of literature on technical writing [9], [10], [11]. But writing of industrial documentation is closely tied with the process of software development. Moreover, industrial and academic texts use different types of software technology for support. Finally, academic texts are more complicated than industrial ones, they have specific approaches to be developed.

There is a large number of books and papers on academic writing, e.g. [12–17]. They focus on text structure and pay special attention to the title [12], introduction [14] and conclusion [16]. There are also works on writing papers on software engineering and computer science [18]. Also there are a lot of papers and books around the writing good thesis (e.g. see [19]). They all, however, concentrate on individual skill training and are addressed to authors. Yet, as it has been said above, students who want to seek a career in non-academic organizations do not require advanced skills in academic writing. What is more desirable for them is an ability to create texts with considerable assistance of their supervisors and other academic staff of the department. Despite a widely spread opinion that students must work independently, their advisors usually play an important role in designing a good text, so it would be reasonable to employ collaborative writing techniques.

These techniques include facilities for synchronizing results of collective writing [20, 21]. There are a number of tools for collective work such as GoogleDoc, ShareLaTeX and others, and various schemes and guidelines for communication, e.g. the author/reader/commentator cycle in the IDEF standards [22]. In [4, 23] mind maps [1] are considered as a collaborative tool to design the structure of academic papers, student theses and software engineering docu-

ments. Nevertheless, the task of writing theses and monitoring student progress has not been addressed yet in terms of managing a package of texts.

A common approach for the task includes: a) classes aimed at improving students' writing skills; b) joint work of students and their supervisors; c) templates, recommendations and requirements of the department/university. As it has already been said, these measures prove to be insufficient.

3. Mind maps and Comapping

Mind Maps were suggested by Tony Buzan in the 1970s to work more efficiently with any kind of information [1]. The idea is to use a simple diagramming notation: the primary (central) object is put in the middle, secondary objects, which clarify the meaning of the central one, are drawn around it and connected with it, etc. (Fig. 2).



Fig. 2. An example of a mind map

This way is suitable for analyzing and understanding any information, especially in case of large volumes. The approach is widely used in education, business, psychology, and other fields. There are also a number of software tools implementing the approach. More information can be found in [1, 23].

Comapping [2] is a collaborative online mind map tool. It supports features such as easy drag and drop, smooth animation, support of large maps with smart auto-focusing features, etc. A tree-like notation (left-to-right mind mapping) is better for computer-based support if combined with a layout algorithm, as it is easier to read and understand than the centre-based one. An example of a mind map in Comapping is shown in Fig. 3.

The tool allows sharing maps among any number of users, as well as notifying them about map changes. These changes are highlighted on the map with detailed information, and other users can review changes and leave comments, which makes Comapping a suitable tool for professors to monitor student works.

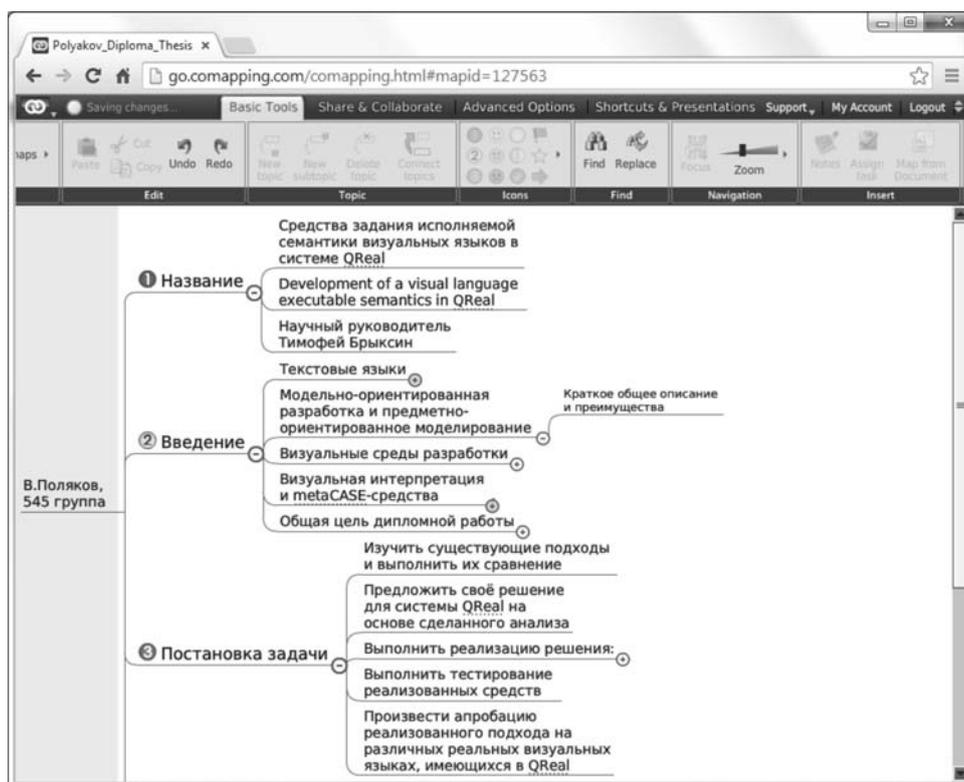


Fig. 3. Example of a mind map diagram in Comapping

4. Fundamentals of the approach

4.1. Text patterns

As it has been mentioned before, we suggest a special course to help students who write theses (final year students) and to monitor the process. Such a course has to be based on a well-defined paper patterns offered to students. These patterns have to be prepared by the department/division of the university that produces graduates, and may vary from one department to another. For example, a thesis on chemistry and one on software engineering will probably have different structures due to different domains.

There is no doubt that an optimal structure can be designed for each thesis paper individually, but that will require a lot of effort. In particular, discussions and adjustments of plans will take a lot of time, which might make the task impossible to complete as part of the course. Additionally, if a single paper pattern is followed, dynamics of the text quality can be measured and, thus, it is possible to assess the method efficiency and suggest ways to improve it.

Our standard paper pattern suggests the following text structure: introduction, research questions, background, main part and conclusion.

- Introduction provides the reader with the statement of research issues. Starting with general trends, it briefly describes all the necessary contexts and explains terms and concepts to be used to set research issues.

- Research questions include a general goal of the thesis and objectives to achieve the goal.

- Background contains all third party results used in the thesis including technologies, problem domain, overall project (if there is one, i.e. thesis is a part of a larger research/industrial project), and also related works.

- Main part is comprised of a number of chapters describing the work done. The structure of the part follows the list of objectives, one chapter for each objective, ideally. It is important that the terminology used in headings is the same as in research questions so that each objective can be easily identified with a chapter describing corresponding results.

- Conclusion contains a list of results, with each result corresponding to an objective, and provides detail on the software tools and platforms employed, developed components, test results, etc.

The pattern may vary depending on the type of thesis (research or industrial).

4.2. Mind mapping

It is noteworthy that the course involves much more than just designing a plan. Reviewing a particular mind map may, for instance, reveal that the student has collected little information about related works or similar solutions/products, or that there are some problems in research questions, or that the thesis results do not correlate with research issues (that means the student has set a problem but does not describe a solution).

Herein we list mind mapping peculiarities essential for the approach.

1. Students should create a mind map for the paper, but not for the entire thesis. Not all of them clearly understand that.

2. Students should create mind maps so that they can write their own texts. Unfortunately, they tend to do the tasks (or to ignore them) simply because the tasks are given by lecturers. It takes some effort to get across the idea to them that the purpose of the course is to help them in a very specific matter, i.e. writing of their thesis papers. Each of them needs to understand how exactly the course can be useful in their own case and try to make their own mind maps for their own thesis paper. It is important that students should feel puzzled, ask questions and delve into the method in order to make the most out of it. In other words, it is important for them to be as pragmatic as possible.

3. The mind map node text should not be informal. In some cases it should be a future chapter/section heading. In other cases the node text is part of the final text (objectives, results, etc.).

4. It is not recommended to use mind maps to write the paper or its parts (sometimes students do it), but to use tools such as Microsoft Word or LaTeX that suit the purpose better.

5. Some students tend to specify too many details in the map. However, a mind map should not be overly elaborate. It should only be sufficient to provide guidelines for writing the text and to demonstrate the supervisor and the course lecturer what is going to be included in the paper and how it is going to be structured.

4.3. Thesis paper development

It is essential that after designing their plans students should move on to writing texts. After the plans have been approved, students have to develop first

drafts. This step often reveals that students misunderstand some aspects of the approach and have to correct their plans and texts. It is important that students do not redesign the entire plan when they start writing. If this does happen, it means the course was of little use for them. Such cases occur but they are very rare (1 or 2 per year).

4.4. Group working process

The approach addresses work with a large group of students (30 to 40) to be monitored by just one lecturer. It can be efficient if an appropriate group work process is established.

The lecturer and the students agree that those of them who would like to receive feedback on their mind maps update them in Comapping before the next scheduled class. The lecturer is notified by email and prepares their comments. The students who have updated the maps are to attend the class to receive the feedback. Neither submitting their updates without afterward coming to class nor coming to class without having made an update comply with the syllabus.

At the beginning, it is better to give feedback by word of mouth. Most students ask multiple questions and some persistently misunderstand some points. All of these are easier to deal with face to face. Later communication can take place online via Comapping.

5. Description of the Course Pattern

5.1 Overview

The presented approach is a learning course template for a group of 30 to 40 students. It is comprised of two parts, namely group work and individual work. During the first part of the course the students attend lectures that familiarize them with the approach. In the second part the lecturer works with each student individually. Due to using mind maps, each student takes no more than 10 to 20 minutes to consult, which allows the lecturer to counsel five to ten students in a 90-minute class. Every student receives in total from four to eight individual sessions during the course.

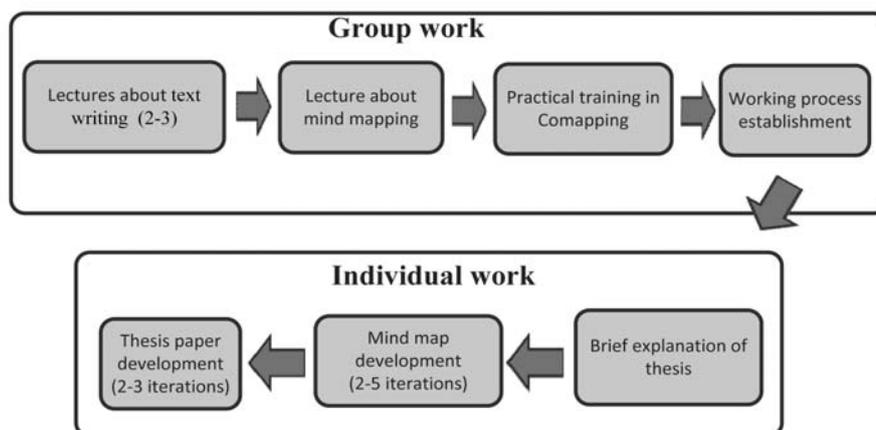


Fig. 4. Course pattern overview

5.2. Course details

Let us now specify every stage of the course.

- **Lectures about text writing.** These lectures ensure that students understand the role of text in thesis development process and become familiar with the recommended thesis paper patterns.

- **Lecture about mind mapping.** Students learn about mind mapping and other types of information visualization techniques [24], about knowledge management [25] and peculiarities of mind mapping.

- **Practical training in Comapping.** Since this software is simple and ergonomic, one lecture is enough, and students go on to work with it independently.

- **Working process establishment.** This is when the work process of the course is explained and agreed on with students. They are also given access to Comapping.

- **Brief explanation of thesis.** The course lecturer is not supposed to go deeply into the details of each thesis because it would take too much effort and time, but he/she should obtain a clear idea of the topic in order to be able to advise on it. With most projects it is not very difficult as they are supervised by our faculty staff and the course lecturer is already familiar with the topic areas (domain-specific modelling [26], random approximation [27], nonlinear control systems [28], etc.). With industrial and third party projects it is slightly more complicated.

- **Mind map development.** The student creates a plan for their paper under the course lecturer's supervision. It usually takes 2 to 5 iterations, depending on how intensively the student works. Ideally, after the first two iterations at the beginning of the seminar the student discusses the plan with their academic supervisor and then corrects it (another two or three iterations).

- **Thesis paper development.** Two or three iterations are optimal for the purpose, as the first attempt normally reveals certain problems (for example, the plan turns out to be inadequate and the student has to go a step back to improve it). The following one or two iterations give the student a chance to correct the errors detected.

5.3. Software engineering specifics

The focus of our approach is on structuring the paper. It is vital for theses on software engineering for the following reasons:

1. There is lack of quality documentation in software industry, which was pointed out by F. Brooks in "The Mythical Man-Month". Almost 40 years later little changed, as it can be seen, for instance, in the documentation of open source projects. Software engineering students take up this trend very soon (especially those with a talent for programming). They are reluctant to learn how to write texts, and while working on their thesis spend more time and make more effort to implement software.

2. Thesis papers usually involve a variety of fields and contexts, which makes writing of introduction and background rather complicated. It is hard to concentrate on just one particular task when a real-life application is being developed.

3. Software engineering does not have a well-established terminology. For example, papers titled "What is software architecture?" [29] are still being published. Besides, developers of software projects use a lot of professional jargon

not only in spoken language but also in writing (email, chat) which significantly deteriorates the style. That is why our approach makes an emphasis on thesis terminology.

6. Evaluation

Fig. 5 shows the statistics of graduates from the department of software engineering, St. Petersburg State University, between 2007 and 2013. The annual average is 39 graduates and most of them end up working in IT industry. Having so many final year students our department needs special tools to provide thesis paper quality (IT specialties are extremely popular among Russian students and our department is the leader at university in terms of the number of students).

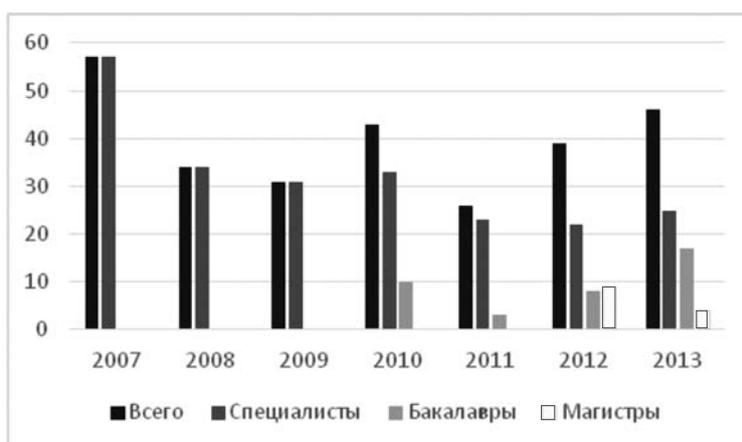


Fig. 5. Statistics of graduates from the department of software engineering, St. Petersburg State University, 2007–2013

We had been assessing the quality of thesis papers at our department for three years before the course was introduced (2007–2009) and for three years after (2011–2013). We used the texts published on the website of the department [22]. We have analyzed only the papers written by students in the “specialist” program, as those in BSc and MSc programs have not graduated in steady batches yet. Most papers are in the same format (Times New Roman, size 14, one and a half line spacing), otherwise they were brought to this format for our purposes. All criteria are assessed on a scale of 1 to 5.

It must be emphasized that we have assessed the quality of the texts themselves, not the entire theses. The criteria we have used are presented in Table 1. They allow evaluating the structure without focusing on semantics (such as whether the theme is fully explored) and their literary strengths and weaknesses can be left aside. Text layout, spelling, etc. are also out of our scope since most of our students are good at these things.

The following criteria have been used to assess the text structure by checking its conformance with the pattern developed by the department. It is important to note that adapting the pattern to individual features of each paper is not an easy task due to diversity of topics and a number of difficulties students face while writing. Moreover, the method does not provide control over the entire text development process. It focuses only on making the plan and creating the first draft, therefore, it is essential to assess final results.

It has to be added that with the use of a single pattern uniform quantitative measurements can be done in the same manner as a defined process enables quantitative process management in CMM. Fig. 6 represents the results of measurements with the abovementioned criteria.

Table 1

Criteria to assess quality of qualification paper texts

№	Criterion	Grade
1.	Quality of introduction	<ul style="list-style-type: none"> • 5 — introduction properly details the topic area, all the contexts are described, information necessary to understand the research questions is provided. • 4 — introduction is too short (we have taken 3 pages for the norm), introduction includes elements of background. • 3 — description of the topic area is not coherent, lacks problem justification, introduction is either too short or too long. • 2 — introduction is replaced with background or research questions. • 1 — absence of introduction.
2.	Quality of research questions and conclusion	<ul style="list-style-type: none"> • 5 — research questions are presented as a set of objectives, which correspond to the structure of the thesis paper and the list of thesis results in conclusion. • 4 — research questions are presented as a set of objectives but they are not properly specified, results in conclusion do not correspond to the objectives, research questions are placed neither at the end of introduction nor in a separate chapter but in some other part of the paper. • 3 — research questions are presented as a set of objectives, and results in conclusion are specified too, but there is no correlation between them and objectives/ paper content, research questions and results are not structured. • 2 — results are not structured and do not correspond to the set objectives; research questions are just outlined, but are not specified as a list of objectives; thesis results are not clearly presented. • 1 — neither objectives nor thesis results are syntactically identifiable, the reader has to find this information in the text.
3.	Quality of background	<ul style="list-style-type: none"> • 5 — background is present as a chapter or a number of chapters where all third party results are discussed. • 4 — there is a background, but some third party results are found in the main part of the paper. • 3 — background is not detailed or complete, third party results are widely present in the main part of the paper. • 2 — background is very short and incomplete, third party results and the employed technologies are mentioned without being thoroughly described. • 1 — background is absent as a chapter, i.e. third party results are either not described at all or presented in the main part of the thesis.
4.	Structure balance	<p>Our analysis has identified the optimal balance of chapters of the thesis paper (without appendices):</p> <ul style="list-style-type: none"> • background should make up from 15% to 35% of the text; • main part should make up 40% to 50% of the text; • title page, table of contents, introduction, research questions, results and conclusions, bibliography — these together should be 15% to 25% of the paper; • appendices cannot to exceed the rest of the paper. <p>If one of the above items is not followed, it leads to taking one point off the grade (the initial grade value for this criterion is five).</p>

D. V. KOZNOV, D. M. NIKOLAEVA. An Approach to Design Thesis Papers in Software Engineering

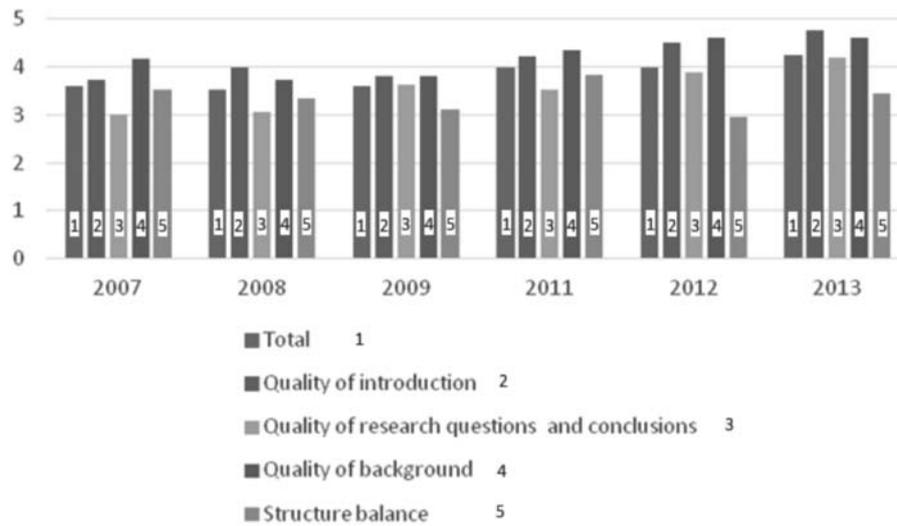


Fig. 6. Statistics of the method implementation

It is evident that the quality of papers have improved after the course was introduced. Before then the average value of the quality characteristic was 3.65, while after it has risen to 4.1. It is noteworthy that the value of ‘structure balance’ has been fluctuating chaotically, which means that this aspect requires more attention.

7. Conclusion

Finally, we shall talk about the problems faced during the course.

- Students tend to underestimate the difficulty of writing texts and are not motivated to create good texts.
- Students display low potential for exploration – they have difficulty in adopting new methods if the latter ones are not immediately obvious.
- Strong stereotypes, such as doing something only because one is told to do or not to do so, because of being highly critical about new information, or to do just enough and no more than to get a pass.
- First draft syndrome: many students are quick in creating the first brief plan of a few general points but it is hard for them to elaborate it. It is often not easy for them to understand that the plan must be thoroughly detailed.
- Involvement of supervisors. Many supervisors are reluctant to use mind maps and are skeptical about any help. This goes along with the Russian academic tradition where a thesis paper is the issue of a student and his or her academic advisor and it is only the final result that is submitted to the department.

In some cases students started to appreciate what they had learnt during the course only after they graduated. One of them, now a director of his own company, has said that the largest benefit of working on his thesis was positive experience in writing a complicated text.

We should also note the limitations to our approach.

- The text of the paper might meet all the criteria mentioned above but not actually be a good text.

- The thesis itself can be of a low quality whereas the text can be quite good. However, during the course the lecturer identifies such thesis and informs the supervisor and the department about problems.
- Our approach is not a method to teach writing. Some students have difficulty in expressing their thoughts on paper, keeping a focus, and following the appropriate text pattern/style. All these are to be taught through different methods/courses.
- This is not a method of thorough work on the text, on its logic, etc. The supervisor still has to read and comment on it, and sometimes to correct it.

REFERENCES

1. R. Meymandpour, J.G. Davis. Ranking Universities Using Linked Open Data. LDOW 2013.
2. Saint Petersburg State University, Software Engineering Department, <http://se.math.spbu.ru/SE/arhiv-diplomnyh-i-kursovyh-rabot>
3. Horizon 2020, Marie Skłodowska-Curie actions, <http://ec.europa.eu/programmes/horizon2020/en/h2020-section/marie-sklodowska-curie-actions>
4. T. Buzan. The mind map book, 2nd edn. BBC Books, London (1995)
5. Comapping, www.comapping.com
6. D. Koznov. Teaching to Write Software Engineering Documents with Focus on Document Design by Means of Mind Maps. International Conference on Computers and Advanced Technology in Education, Napoli, Italy, June 25–27 2012. (CATE2012).
7. T. T. Barker. Writing software documentation: A task-oriented approach (Part of the Allyn & Bacon Series in Technical Communication, Longman, 2002).
8. E. H. Weiss. How To Write Usable User Documentation. 2nd edition, Phoenix: Oryx Press, 1991.
9. J. Williams. Transformations in technical communication pedagogy: engineering, writing, and the ABET engineering criteria. SIGDOC 2000. P.75–79.
10. H. Wright. Technical Writing Tools for Engineers and Scientists. Computing in Science and Engineering (CSE) 12(5), 2010. P. 98–103.
11. L. Seawright. Teaching Technical Writing: Opportunities for International Collaboration. iJEP 4(2), 2014. P. 28–30.
12. T. M. Annesley. The Title Says It All. Clinical Chemistry. March 2010; 56 (3). P. 357–360.
13. T. M. Annesley. The Abstract and the Elevator Talk: A Tale of Two Summaries. Clinical Chemistry. April 2010; 56 (4). P. 521–524.
14. T. M. Annesley. “It was a cold and rainy night”: Set the Scene with a Good Introduction. Clinical Chemistry May 2010; 56 (5). P. 708–713.
15. T. M. Annesley. Who, What, When, Where, How, and Why: The Ingredients in the Recipe for a Successful Methods. Section Clinical Chemistry June 2010; 56 (6). P. 897–901.
16. T. M. Annesley. Show Your Cards: The Results Section and the Poker Game Clinical Chemistry July 2010; 56 (7). P. 1066–1070.
17. T. M. Annesley. Giving Credit: Citations and References. Clinical Chemistry January 57(1) (2011). P. 14–17.
18. I. Stojmenovic. Editor’s Note: How to Write Research Articles in Computing and Engineering Disciplines. IEEE Trans. Parallel Distrib. Syst. (TPDS) 21(2). P. 145–147 (2010).
19. D. Evans, P. Gruba, J. Zobel. How to Write a Better Thesis. Springer, 2014.
20. T. Lindholm. A three-way merge for XML documents. ACM Symposium on Document Engineering 2004. P. 1–10.

21. G. Oster, H. Skaf-Molli, P. Molli, H. Naja-Jazzar: Supporting Collaborative Writing of XML Documents. ICEIS 2007. P. 335-341.
22. Integration Definition For Function Modeling (IDEF0). Draft Federal Information Processing Standards Publication 183, 1993, 79 p.
23. D. Koznov, M. Pliskin. Computer-Supported Collaborative Learning with Mind-Maps. T. Margaria and B. Steffen (Eds.): ISoLA 2008, CCIS Vol. 17, 2008. Springer-Verlag, Berlin Heidelberg, 2008. P. 478-489.
24. D. Kudryavtsev, T. Gavrilova, I. Leshcheva. One approach to the classification of business knowledge diagrams: Practical view. Federated Conference on Computer Science and Information Systems, FedCSIS 2013. P. 1259-1265.
25. T. Gavrilova. Knowledge engineering for non-engineers. 6th IFIP International Conference on Intelligent Information Processing, IIP 2010; Manchester; United Kingdom; 13 October 2010 through 16 October 2010. P. 225-233.
26. A. Kuzenkova, A. Deripaska, T. Bryksin, Y. Litvinov, V. Polyakov. QReal DSM platform: An environment for creation of specific visual IDEs. 8th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2013; Angers; France; 4 July 2013 through 6 July 2013. P. 205-211.
27. O. Granichin, L. Gurevich, A. Vakhitov. Discrete-time minimum tracking based on stochastic approximation algorithm with randomized differences. 48th IEEE Conference on Decision and Control held jointly with 2009 28th Chinese Control Conference, CDC/CCC 2009; Shanghai; China; 15 December 2009 through 18 December 2009. P. 5763-5767.
28. G. Leonov, N. Kuznetsov. Analytical-numerical methods for investigation of hidden oscillations in nonlinear control systems. IFAC Proceedings Volumes (IFAC PapersOnline)18 (PART 1). P. 2494-2505.
29. F. Solms. What is software architecture? SAICSIT Conf. 2012. P. 363-373.

Кознов Дмитрий Владимирович

Кандидат физико-математических наук.

Научная отрасль:

Санкт-Петербургский государственный университет,
доцент кафедры системного программирования.

E-mail: dkoznov@yandex.ru

Николаева Дарья Михайловна

Научная отрасль:

Санкт-Петербургский государственный университет,
студент кафедры ??

E-mail: daryanika@gmail.com