

Russian Federation Government
Federal State Autonomous Educational Institution of Higher Education
National Research University
"Higher School of Economics"
School of Software Engineering

APPROVED by
Head of the Master Program
"System and Software Engineering"
Professor of School of Software Engineering,
Dr.Tech.Sci.

_____ Dmitry V. Alexandrov
« ___ » _____ 201_ г.

Course work

Field of study
09.04.04 – Software Engineering

Scientific Advisor _____ position, place of work _____ academic degree, academic title _____ name _____ grade _____ signature, date	Author _____ group student Master Program "System and Software Engineering" _____ name _____ signature, date
--	--

Moscow, 201_

Course work recommendations

The course work should be between 20 and 30 pages in length, excluding appendices and illustrations. It could be a mini dissertation and should present the key aspects of the project work, a distillation of the results, and a suitable analysis.

Subject

Students may choose any topic that will allow them to demonstrate understanding of software engineering through the medium of a course work. The best way of doing this is usually to choose a topic related to one or more of the subjects taught on the Programme. Many students choose topics that are related to their current employment, allowing them to see how the ideas taught in the Programme may be applied alongside current practice.

There are three broad categories:

1 Product-based

- involving the application of software engineering techniques to the specification, analysis, design, development, or testing of a particular product. This product could be entirely fictitious, it could exist only as a part of another system, it could be hardware or software, but it must afford an opportunity to apply software engineering techniques. Any technique or combination of techniques could be applied to—or around—this product. There might be a study of requirements, a security analysis, one or more specifications, some design work, some analysis, some programming, or some testing. Examples include: an object-oriented design and implementation, with some accompanying analysis; the formal specification of a software component; a case study in requirements engineering.

2 Theory-based

- involving the extension of a particular technique, or a combination of techniques, for use with a particular class of application.

3 Process-based

- involving an examination or assessment of particular techniques, their development and application, in academia or in industry. If no application is involved, the resulting course work must contain enough original insight, analysis, or critical reflection to demonstrate the student's own understanding. Examples include: a description and critical analysis of the software testing process for a particular product line; an analysis of a software standard, with proposals for change, based on ideas from the Programme.

Each type of project has its own advantages: the development or description of a particular product might make it easier to define goals, or to measure progress; a theory-based project is unlikely to suffer from implementation problems; a study of process might result in immediate improvements to the working environment.

Proposal

The work starts with a project proposal containing:

- 1 a description of the subject matter: product, theory, application area, or problem domain;
- 2 a brief account of the original contribution that the project work might be expected to make;

- 3 a plan, or outline, of the course work, explaining how it will demonstrate the student's understanding;
- 4 a list of resources that will be required.
- 5 a suggested schedule of tasks and delivery dates; and
- 6 an outline of any social, legal, or ethical issues relating to the work.

While working on their project and course work, the student should take care to keep their supervisor informed of progress. The primary mode of communication will be email. As the work progresses, there might be considerable variation from the original proposal: any drastic change may require approval.

Course work

The results of the course work are presented in a short paper, of 20-30 pages. The main matter of the course work will contain between three and eight sections, beginning with an introduction and ending with a discussion. The front matter is what comes beforehand: the cover page, the abstract, the table of contents. The back matter is what comes afterwards: the bibliography, and any appendices.

Front matter

The student's name, and the name of their university, should appear on the cover of the course work, along with the title and the year of submission.

The second page should include an abstract of between 100 and 200 words: this should be written in the passive voice. The abstract should avoid mention of any company-specific information.

The last part of the front matter is the table of contents, which should list the name of each top-level or second-level section, together with the number of the page where it starts. There is no need to include third- or fourth-level sections in the table of contents.

Main matter

The main matter of the course work should be divided into either chapters or sections, with lower levels of sectioning as necessary. In a document of this length, more than two levels of numbered sectioning is likely to inhibit readability. The following is a generic structure describing how a course work might be organised:

1 Introduction (1-2 pages)

The first section should do three things: introduce the subject matter: explaining the motivation for the project, and saying something about the background area; describe the contribution made by the course work: the results of the project, the impact or value of the work; provide an outline of the course work, explaining how this contribution is realised in the subsequent sections.

This section may be only a page or two in length, but it is the most important part of the course work. It will shape the expectations of the reader, and provide them with a guide to the rest of the work.

2 Application area (2-7 pages)

In most cases, there will be a second section that describes the context of the work: the application area, the problem domain, or the theory that is to be extended or analysed. In other cases, if this description is only a couple of pages long, it could be made part of the Introduction.

This description should be limited to the context, not the work itself. If the project involves the creation of a model for a software product, then this section should describe the product, but not the model. If the project is theory-based, this description should describe the existing theory, but not the proposed extensions.

As with the Introduction, this section will shape the reader's expectations. It should provide enough information about the context to allow the reader to assess the contribution made. It should mention only those aspects or features that are relevant to the subsequent sections, emphasising those that are particularly relevant.

3 Project work (15-25 pages)

The core of the course work: one or more sections presenting the results of the project work: If the project is *product-based*, there might be sections on requirements capture and analysis, specification, design, implementation, or testing. There is no need to address all of these areas. A course work might present three different specifications of a simple product or component, and a subsequent implementation, in four successive sections: State-based Specification (4 pages); Behavioural Specification (3 pages); UML Model (4 pages); code Implementation (4 pages).

Here, a commentary might be expected: something that compares and contrasts the different specifications, and explains how they might be used to validate, or even generate, the implementation. However, it would not then be necessary to include an account of testing strategies or requirements analysis, nor a detailed analysis of the consequences of the design decisions associated with the specifications and implementation.

Alternatively, a course work might present a single specification, with a formal or rigorous account of properties and perhaps their verification. If the specification and the properties are sufficiently complex or novel, this might comprise the whole of the project work.

Alternatively, there might be another section describing a prototype or implementation.

Thus we might see: Formal Specification (6 pages); Analysis (5 pages); Prototype (4 pages). Finally, there might be two models, each of which has its own analysis and implementation, repeating this structure one more time.

If the project is *theory-based* then this part of the course work could be broken in sections describing different aspects of the overall achievement.

The same applies to projects that are process-based: this part of the course work could be broken into sections describing different issues addressed. For example, a course work that contains a critical analysis of a particular company's development process might, after a section that describes that process, present the work as: Requirements Engineering (5 pages); Development Management (4 pages); Testing Strategy (3 pages); Proposals for Change (3 pages).

4 Discussion (3-6 pages)

The preceding sections should contain a certain amount of reflection and analysis. However, anything particularly insightful should be repeated here, and discussed in as much detail as seems appropriate.

In most cases, the discussion section will be one or two pages. However, it may be that the student has learnt or achieved a great deal that is not immediately apparent from the preceding chapters. If that is the case, then a deeper discussion may be necessary.

If the specification now seems inadequate, or the implementation has been unsuccessful, the student could use an extended discussion to explain exactly what the problems are, and to suggest how things might be remedied. This could be as informative, and rewarding, as a reworked project: it might also be the only way to produce a satisfactory course work in the time available.

The discussion might also complement the introduction by relating the results of the project work back to the original context, suggesting further work or comparing the student's work with that of others.

The page counts shown above are simply for guidance. Like the numbering of sections, or the proportion of illustrations, or the amount of code displayed, they will vary quite widely from course work to course work.

Back matter

The back matter should begin with a bibliography, or a list of references. In a product-based project, this might take up half a page; in a theory- or process-based project, it might be longer. Either way, it is essential: it provides a way for the reader to learn more about the context of the work.

Many course works have an appendix of some sort. This may contain excerpts from the code of an implementation, or screenshots of an application in use, or a mathematical proof supporting the conclusions of a specification or analysis section. Any appendix will be excluded when the formal extent of the course work is measured; however, you should note that an over-long appendix might be ignored by the Examiners.

Presentation and submission

There is no need for doublespaced text: the default line spacing used by most packages is perfectly adequate.

The course work should be printed using one side of each sheet of paper. There is no fixed format or document template, and either A4 or letter paper is acceptable. As a rough guide, you should expect to find no more than 10 or 12 words per line, and no more than 40-50 lines per page. Each page, apart from the cover, should be numbered. It may also be appropriate to add a header to each page, identifying the current top-level section.

Upcoming submission deadlines are published on the website of the program.
