Dissertation Writing – CS & PBA WEB

By Per S. Laursen, EASJ

v. 0.3 – August 2013
0.4 - April 2014

Table of Content

Preface

This document is intended for students at the below two educations at EASJ (ZIBAT):

- Advanced Programme in Computer Science (called **CS**). Note that there is a Danish version of this education (Datamatiker), and an international version.
- Bachelor in Web Development (called **PBA WEB**)

The document is supposed to be able to answer many of those questions a student might have with regards to writing a final dissertation for these educations. Note that information about the dissertation work can also be found at the sources below (the schools official "manuals" for dissertation work). Note that the manuals may be slightly outdated

- Manual for CS dissertation work
- Manual for PBA WEB dissertation work

This document is thus supposed to supplement (not replace) the above sources, in particular with regards to what to actually write in a dissertation, and how to achieve this.

Introduction

Writing the final dissertation can – depending on your previous experiences – seem like an very large and difficult task… and it is! It is the place where you should demonstrate that you are able to apply the knowledge you have gained during the study to solve a complex problem. That is indeed a demanding job, and will most likely give rise to many questions. This document tries to answer many of those questions, and provide some guidelines as to how to progress through the work. The document will probably not be able to answer all of your questions, but hopefully it can at least de-mystify what it is all about.

What to find in this document

The document contains the below sections (click the links to proceed to the desired section):

Prerequisites for the dissertation

Since the dissertation is the final leg of the education, you are formally not allowed to attend the dissertation examination (see later) before

- All other exams in the education have been passed
- Your internship has been approved.

Of course, nobody can prevent you from starting to think about your dissertation earlier on.

However, note that you cannot receive specific supervision on your dissertation work before your Internship Report has been approved. In practice, the supervisor will be available for e.g. discussing possible dissertation topics before the Internship has been formally approved, but detailed supervision requires that the Internship has been approved.

Another prerequisite for the dissertation is that the **Dissertation Project Contract** must be submitted to the school AND must have been approved by the supervisor. The contract contains the project title, a brief project description, and a few formalities. A template for the project contract is found at the end of the official project manuals (follow the links in the Preface). The topic for your dissertation can – and should – of course be discussed with your supervisor before submitting the project contract (see the chapter ***Choosing a topic***).

Practicalities

This section is intended to answer many of those practical questions that arise in relation to the dissertation work.

Can I write a dissertation alone, or must it be done in a group?

Legal group sizes for dissertation work are from 1 (one) to 4 (four) students. So yes, you can write a dissertation alone, and in practice, most students do this. However, the school encourages students to write dissertations in groups. Being in a group usually makes the task easier, since you have much less risk of being stuck during the work. Working alone on a project of this size does require a lot of self-dicipline…

How much time do I have for writing the dissertation?

Formally, the time period dedicated to dissertation work is the time from the completion of the Internship until the deadline for handing in the dissertation. With small variations – depending on the semester and the details of the Internship – this period is about 10-11 weeks.

How much work am I supposed to put into the dissertation work?

The dissertation is defined as being equivalent to 15 ECTS points. Since 1 ECTS point is supposed to represent a work effort of about 27 hours, this gives a total of about 400 hours of work. Since the time period is typically 10-11 weeks, working on the dissertation is indeed a full-time job, requiring about 40 hours of work per week during that period.

Can I work abroad, e.g. in my home country?

There are no requirements concerning your whereabouts during the dissertation work period. Of course, being abroad will put some limitations on the given supervision, but it is of course possible to receive supervision though e-mail, Skype, etc.. You should discuss and arrange this with your supervisor.

How many pages are required?

Formally, there is only an upper limit on the page size:

- For a one-student group: 40 pages
- For a two-student group: 60 pages
- For a three-student group: 80 pages
- For a four-student group: 100 pages

Note that this page count does not include material in appendices. In practice, many students greatest concern seems to be if they have written too few pages… So, is there a practical lower limit? That is very hard to answer generally. Instead of focusing too much on page count, you should rather concentrate on convincing the reader that you have spent enough time on the project, and that you have spent the time wisely. On the other hand, delivering the maximal number of pages is not a guarantee for getting a high mark for your dissertation.

In relation to page counts, you should also be aware that a "page" in this context in not always equal to one physical page. A "page" – or more formally, a so-called "normal page" – is defined as containing 2400 characters, including space and other special characters. This means that playing around with font sizes, line spacings and page breaks does not change the page count, since it is formally defined as the total number of characters in the report, divided by 2400.

A very natural question is then "what about illustrations"? Formally, any graphics, figures and code listings are not counted as characters, and therefore do not contribute to the page count. This is actually only to your advantage, since the risk of hitting the page limit just gets lower.

With modern report authoring tools, it is usually not a problem to determine the exact character count. In Microsoft Word, the count is readily available in the status bar.

The dissertation supervisor

Since working on a dissertation is a large and complex task, a supervisor is assigned to each dissertation project. The supervisor will be a teacher from the education. You can influence who will be your supervisor; sometime during the semester before the internship/dissertation semester, you will be asked to submit a prioritised list of preferred supervisors (if any) – the school will then try to assign one of your preferred supervisors to your project. There is however no guarantee that you will get your top priority, since the assignment of supervisors to projects also depends on the other duties a teacher may have during the semester in question

Who should I choose as my preferred supervisors?

All teachers at the school are fully competent supervisors. So choosing one over another is usually driven by your perception of a teacher – is it a person that you feel comfortable with, and can cooperate with. Also, you should be aware that different teachers of course have different areas of expertise. Some teachers are primarily oriented towards technological topics like programming, technology frameworks, etc., while other teachers are primarily oriented towards e.g. business, design, usability, strategy, etc.. All teachers can provide general supervision about project planning, structure, methodology, etc., but the "depth" of supervision is of course limited by the teachers competences. A teacher who specialises in programming may not be able to provide "deep" supervision about design and digital aesthetics, and vice versa. So take this into consideration when choosing supervisors.

How much supervision can I get?

The usual procedure for supervision is to arrange a weekly or bi-weekly meeting, where the student(s) can submit material to the supervisor some days in advance, if they wish to receive comments to some specific material. Other arrangements can of course be made, if e.g. the project is done abroad. In any case, the supervisor is not an unlimited resource, so use the supervisor wisely. As a rule of thumb, the supervisor will have time for about four hours of meeting activity (per student), plus the time needed by the supervisor to prepare for the meetings. Some e-mail correspondence will usually also be part of the supervision, but there will also be limits to this. You cannot just send e-mails *ad libitum* to your supervisor, and expect to get answers the next day. You should also try to use your supervisor in a fairly "regular" manner during the project. We sometimes see students that do not really seek supervision before very late (maybe a week before the deadline), and then floods the supervisor with material the last few days. This is not going to work; you cannot expect to monopolise the supervisor's time during the last few days of the project.

How does supervision work?

In principle, you can ask the supervisor about virtually anything during the project period. You can have the supervisor review and comment on drafts, help you find materials, etc.. BUT – and this is a very important point – the initiative must come from yourself! Given the limits to supervision discussed above, you should

also utilise the supervisor in a relevant way. By that is meant that asking the supervisor to spend time correcting grammatical errors in your drafts is probably a bad way to use the supervisor, since many others could do this. Use the supervisor for matters relating to project structure, methodology, finding relevant material, advice on domain-specific issues, etc..

Is it the responsibility of the supervisor that I get a good mark?

No. The supervisor will not just take you by the hand and guide you to a top mark! If this could be done with all students, then why do a project at all? Remember, the dissertation work is an examination, so the supervisor cannot just give you a complete answer to everything. The supervisor can help you choose the right tools for the job, find materials to work with, help you plan your work, but in the end, it is you that have to apply the tools. An analogy could be a painting; the supervisor can help you pick proper paint and canvas, choose a good motive, discuss color theory, etc. with you, but that is in no way a guarantee that you will end up with a piece of art, since you will have to do the actual painting.

Can I change my supervisor during the project?

In principle, yes. However, it is definitely not something we encourage. First, it might simply be impossible due to other teachers duties. Second, consider if the problem really lies with the supervisor, or if it is your own performance that lacks something. Changing the supervisor as a consequence of your own poor performance will seldom result in anything positive.

Choosing a topic

It is definitely not easy to come up with a good topic for your dissertation work. In some schools, the students are given a large "buffet" of topics to pick from – this is not the case here…

Who determines the topic for the dissertation?

In the official dissertation manuals (as found through the links in the Preface), it is stated about the dissertation topic: ***"The topic for the final project is determined by the student in cooperation with the institution and a company, if possible. The institution approves the topic of the project"***. So, the main responsibility for defining a topic rests with the student! Of course, the school (more specifically your supervisor) will assist with this, but the initiative must come from the student.

Are there any limits to what the topic can be?

Yes, but they are somewhat weakly defined. The only actual requirement is that the topic can be associated with one or more of the courses on the education. Writing a dissertation on the Roman Empire will thus most likely be hard to link to a course, but as long as the topic is within the realm of Web Development and/or Computer Science, there is quite a lot of freedom.

To give an impression of the diversity of possible topics, here are some topics of existing dissertations:

- Developing a website using a PHP-based open-source framework (PBA WEB)
- Research study of icon design for websites (PBA WEB)
- Defining a strategy for an educational institution for using social media in marketing (PBA WEB)
- Comparing two Windows GUI technologies (CS)
- Exploring the Micsosoft Silverlight technology (CS)
- Developing software for remote maintenance of computers (CS)

How do I get an idea for a topic?

The primary driver for choosing a topic should be personal interest. It is much easier to keep up motivation if you feel you are working on an interesting topic. However, it may not be all that have a burning interest in a particular topic…then what?

One way of generating ideas for a topics is to review the curriculum for the courses you have taken. Maybe there were some topics that you thought were more interesting than others, and where you even wanted to know more than what was taught in class. That process could narrow down your search to a few topics. Then you could also try to think about a domain where you would like to know more. It could maybe be things as diverse as data visualisation, games, tools for disabled people, production planning, etc.. Ideally, that should give you some ideas for working with certain topics within certain domains. Given that, you could maybe start to search for more information online, to give you some more concrete ideas. Of course,

your supervisor will be available for discussions about such ideas, before you make a final choice. The final choice must also be approved by the school, so in that sense, a supervisor can still turn down an idea, if it is found to be maybe too thin, too complex, too far away from the education topics, etc..

Another way – which is often quite attractive – is to define a project which is somehow related to your Internship work. An Internship can often end up with a project which is only partially completed, and it is perfectly acceptable to pick up such a project as the dissertation topic. That gives the advantages of already knowing about the domain, tools, techniques, etc., and can thus give you a head start on the project. Of course the boundaries between the Internship work and the dissertation work must then be clearly established, since you cannot as such get credit for your Internship work in the dissertation. But if those boundaries can be established, you can have a perfectly valid project.

If none of the above can help you come up with an idea, you could also have a look at some existing dissertations. Your supervisor can arrange that for you. Just be aware that just choosing a topic identical or very similar to an existing dissertation is probably not acceptable. The existing dissertations should only be used for inspiration.

Types of projects

Before proceeding further with the project, it is useful to first consider what type of project it is supposed to be. Different project types require different methodologies (we discuss methodologies further in a later chapter), so being sure of the type is essential for choosing an appropriate methodology. Below follows a list of possible project types. Note that the list does not give a lot of details about the project types – we elaborate on the individual project types in the **Methodology** chapter.

- **Software product development** – at least in Computer Science, this is the "classic" project type concerning development of a software application, using a well-defined development process to implement a number of requirements.
- **Website design and development** – also a typical project type, where the focus is on the website content and functionality. Depending on the specific project, the weight can be towards design or implementation. Pure website design projects without actual implementation are also possible.
- **Technology exploration/comparison** – a project type aiming specifically at exploring the properties and benefits of applying a specific technology within a relevant realm, and comparing the technology to another technology within the same problem realm. This could e.g. be using a certain framework for web development, using graphics class libraries, etc..
- **Technology integration** – some projects deal with tying together many pieces of technology, which in itself can pose many challenges. This could be in a setting with many different hardware and/or software components that need to interact.
- **Research study** – a more "classic" academic-type project, where e.g. a technology or technique is researched thoroughly, not necessarily including any actual development. The goal could e.g. be to give a thorough overview of the current status of, say, a development technology.
- **Media strategy** – a project type dealing more with the business side of using certain media, for instance social media for marketing and customer interaction. Could e.g. be to develop web media strategies for an organisation.

Some of the project types are overlapping, and there might also be projects that cannot fit into any of these categories. This is not necessarily a problem; the types are mainly meant as a guide for choosing a proper methodology, and if a project does not fit, the methodology will have to be developed from scratch. This may take more effort, but is of course possible.

Again, the reason to do this categorisation is to make sure we know what the project should focus on, and thereby choose a proper methodology. Consider this project: ***Development of a website using a open-source technology framework***. Is this a "Website design and development"-project, or is it a "Technology exploration/comparison"-project? In the first case, the focus is much more on the website as such (design, functionality, target audience, usability, etc.), with the framework just being a convenient tool, that is not so interesting in itself. In the second case, the focus will be much more on what the framework can do for us (enforce compliance with design or architecture patterns, auto-generate functionality, manage persistence, etc.), while the website itself is just a setting for exploring the framework, and not so interesting in itself. Making this distinction will obviously affect the project very significantly, and it should therefore be made at a very early stage.

Problem formulation

It might sound very academic to have to explicitly define a “problem” to work on. First, the term “problem” is to be understood in a broad sense. It is not something like a previously unresolved mathematical proof; it is just that specific challenge we are facing in this project. Examples of problems formulations could be:

- “How can organisation X use Facebook as a platform for improving sales and customer relations?”
- “How can we implement a software application for (…) using the technologies C#, SQL Server and JavaScript?”
- “What are the main benefits and drawbacks of using the technology framework X for developing a web-based collaboration application?”
- “How can we implement an application for remote managent of computers in a company computer network?”
- “What is the current trend in web server software with regards to (…)?”

These problem formulations are somewhat simplistic, but it is just to illustrate that there is nothing mystical about a problem formulation. But still, if no problem formulation is stated, it becomes unclear what the goal of the project actually is. In fact, the problem formulation is so important that it must be approved by the school.

An important role of the problem formulation is related to the conclusion. The conclusion should – among other things – explicitly answer the question(s)/problems(s) stated in the problem formulation. We discuss how to write conclusions later.

Looking at the above, it might look like the problem formulation will be a very short chapter. It usually will be. Note, however, that you should supplement your problem formulation with relevant discussions about why this is an interesting question/problem at all, why it is complex enough to form the basis for a dissertation, if possible break the main question/problem into subparts, etc.. Remember, the clearer and more detailed you can make your problem formulation, the more guidance you will have on how to proceed with the actual work.

Methodology

The choice and definition of methodology is crucial for the project. The methodology is the driving force of the project; it decides what activities you should perform during the project, what conclusions you can draw, etc.. Without a properly chosen methodology, the project can easily lose focus, and you may end up performing activities that were not really needed, while maybe missing out on other essential activities.

There is obviously a very close relationship between the problem formulation and the methodology. The whole purpose of the dissertation is to provide answers/solutions to the questions/problems that were stated in the problem formulation. The methodology defines how you are going to achieve that; activities you will have to perform, "artifacts" that will be the result of your activities, techniques for drawing the final conclusions, etc.. With a well-defined methodology, there is never any doubt about what to do next, or if an activity is relevant or not.

Since there is no detailed methodology that fits all project types, a methodology that matches the chosen project type must be chosen and adapted to the specific project. This is why we defined the project types above, so we can now present methodology outlines appriopriate for these types. Note, however, that this is indeed only outlines; you will still need to flesh out the methodology for your specific project.

In the subsequent sections will follow methodology outlines for each of the project types mentioned previously:

Software product development

A project of this type will deal with the development of a software application, where the focus will be on the functionality rather than the technology. That is, the chosen tools might be tools that are well-known from the education (since the tools themselves are not at the center of attention here), but the application itself must contain a certain complexity with regards to its functionality. The challenge in such a project will thus be to define, refine and implement functional requirements for the application. For this purpose, a typical choice of methodology will be to apply an existing and well-described software development process, be it Unified Process, Agile Development, etc.. The specific choice will of course depend on the nature of the project. The artifacts produced during such a project could be – besides the application itself – business analyses, designs at various levels, test specifications, descriptions of select parts of the code, GUI designs etc.. Test cases, user feedback etc. can be used for determining whether the "problem" for such a project has actually been solved

Website design and development

For website design and development, it is quite possible to use an approach similar to the approach described for software product development, if the focus is primarily on the functional aspects of the website. There might be different areas to focus on, however, like information architecture, search engine optimisation, division of responsibilities between client and server, etc.. If the focus is more towards, say, design issues, usability, etc.., it may be more relevant to employ tools like GUI prototyping, usability studies, user focus groups, etc.. The specific artifacts could here be the GUI prototypes themselves (wire-

frame, high-detail), transcripts and analyses from usability studies, and also more traditional artifacts like test cases and test results.

Technology exploration/comparison

A technology exploration/comparison project can appear to be very similar to e.g. a software product development project. However, the important difference is that the focus is not as such on the developed application (the application might only reach a prototype stage); it is on the technology that is being used for the task. This means that the developed "product" (application/website/…) does not necessarily need to be particularly complex, since it serves as a setting for exploring the technology. Conversely, the technology itself should then be interesting and complex enough to serve as the main topic for a dissertation. Examples could be a Content Management System or a software development framework. Instead of the more traditional artifacts like requirements and designs, the artifacts would typically be detailed descriptions on how to apply the technology during key stages of development, at a level where it is clearly demonstrated what makes the investigated technology interesting and worthwhile. Assessing whether it is "worthwhile" to apply a certain technology indicates some kind of comparison. This will almost always be a part of such projects. The technology under investigation should be held up against a different approach to solving the same problems, and the challenge is then to demonstrate that the applied technology is indeed beneficial. This could be in terms of reduced complexity, fewer lines of code, shorter development time, improved usability, etc.. Defining exactly what makes a certain technology "worthwhile" to use will be a very important element in such a project.

Technology integration

A project like this could involve an atypical and complex technological "setup" for the project, for instance involving unusual hardware platforms, integration to external systems in a company, etc.. The challenge for such situations can be related to connectivity, i.e. simpy enabling the various parts of the envisioned system to communicate. It might be that the system under development needs to extract data from an existing, external system, needs to use web services, etc.. Integrating certain technologies should of course have a purpose in itself, so there will probably be an element of software application or website development in such a project as well, as there might also be a need to explore and describe certain technolgies used for the integration. In that sense, this type of project is similar to the already described types, but since the integration of the technological components themselves is supposedly complex, this activity can be allowed to take the center stage, while the effort dedicated to more traditional activities for software or website development can be toned down. The artifacts in such a project could be detailed descriptions of the application of the integrating technology, reflections over their usefulness and complexity, descriptions of shortcomings, etc.. These artifacts will thus be similar to those produced during technology exploration, but there may not be any obvious alternative technology to compare against.

Research study

This type of project is probably closer to what could be called a traditional academic study or survey. The purpose of such a project could be to investigate a topic (a technology, a development process, etc.) from a more theoretical viewpoint, and may therefore not necessarily involve development of a product as such. An example could be to research the current state of, say, .NET technology (this is only meant as a hypothetical example). Such a study would then involve gathering information from external sources to shed light on those particular properties that are found interesting. In the example, it could e.g. be trends in usage in industries, statistics for jobs advestiments, trends for competing technologies, and so forth. Since there is typically no product development involved, one should be prepared to put in a significant effort in the information gathering, analysis, source critique and similar research-oriented activities, since they will be the primary activities in the project. If not, one may easily end up with a project that will be deemed too "light" for a dissertation. An important part of such a project will be a reflective part, containing ones own appreciation of the obtained results. It will not be sufficient just to gather and describe the obtained data.

Media strategy

A project in this category will have a more business-oriented perspective. An example could be to develop a strategy for using social media for an organisation, e.g. for improving relations with customers, to increase public awareness of the organisation, etc.. Artifacts in such a project could be business strategies for using the particular media, organisational analyses (how will the strategy affect the organisation), marketing strategies, target audience and/or personae definitions, and also more concrete artifacts as e.g. prototypes for websites, Facebook profiles, material for other media types, etc.. Performing such a project will require rather closer cooperation with an actual organisation, since continuous feedback from the organisation is essential. A project like this cannot be a purely theoretical project.

Report structure

Once the topic for the dissertation has been found, a problem formulation has been completed, and a suitable methodology has been chosen and defined, the actual work on "processing" the problem begins. As you can hopefully tell from the outlines of methodology above, the specific "artifacts" that will result from the work can be very different from project to project. It is therefore not possible to give a general and complete description of what a report should contain. In fact, figuring out the exact set of needed artifacts and the resulting report structure is also an important part of your dissertation work.

However, no matter the specific topic for your dissertation work, your project should follow a outline as given below.

- A **front page**, including: project title, name of group members, the name of the education, the name of the supervisor, and the project period. It is a formal requirement that your report includes such a front page, so remember it!
- An **abstract**, which is a short (at most half a page) summary of the project, including the main problem formulation, the main features of the methodology, and the main conclusions. Note that an abstract should be self-contained, in the sense that you should not leave anything out of the e.g. the introduction, with the argument that you have already written it in the abstract. You typically write the abstract as the last thing, once all others parts are done.
- A **Table of Contents**, including page numbers
- An **introduction**, where you introduce the problem, the context in which the problem is solved, etc. If you are doing the project in collaboration with a company, this would be the place to include a short description of the company, its business, etc.. If the project deals with a special domain, it would also be appropriate to introduce this domain in the Introduction. You can also introduce special terms, etc.. The Introduction should enable the reader to understand the upcoming problem formulation.
- A **problem formulation**, where you state explicitly what questions/problems it is that this project will specifically address. See the chapter ***Problem Formulation*** for further details.
- A **methodology**, where you formulate the details of the methodology you will use to address the questions/problems from the problem formulation. See the chapter ***Methodology*** for further details.
- **All the sections needed to describe the produced artifacts**. This will be the bulk of the report. If you need to do a Business Analysis, there will a chapter on that. If you need to do Software Design, there will be a chapter on that, etc.. Again, we cannot be more specific about what this part of the report will contain, since it is highly dependent on the chosen methodology.
- A **conclusion**, where the main achievements and learning points are discussed in relation to the problem formulation. See the chapter **Writing a Conclusion**, which follows below.
- A **bibliography**, with references to all the materials you have used. This can be books, websites, videos, etc.. It is indeed important to mention all sources, in particular if you have quoted from the source in the report text. Not mentioning a source which you have quoted from can formally be considered cheating!
- A **list of appendices**, with a short explanation of the purpose and content of each appendix
- The **appendices** themselves, if you have any

Writing a Conclusion

The conclusion should be the culmination of your report. It is here you should describe how well you succeeded in solving the problems/questions that were stated in the problem formulation. This is a very important point – there should be a certain "symmetry" between the problem formulation and the conclusion. If you have mentioned some problems/questions in the problem formulation, you should refer to them in the conclusion. A reader should be able to just read the problem formulation and the conclusion, and from that alone decide if it will be worthwhile to read the entire report, if the reader is looking for an answer to some specific issue. A conclusion should therefore be taken seriously; avoid writing a bland, generic conclusion like *"It was a very interesting project, and I learned quite a lot…"*. It has to be much more specific. If you claim to have solved a problem, you should explicitly refer back to those parts of the report where you demonstrate this.

A conclusion can also be used to "look ahead" and put things in perspective. Depending on your project type, there might be certain tasks that would still remain to be done, if this was a real-life project. If you have developed a website, you could discuss what should happen in terms of deployment, monitoring, maintenance, etc.. If you have explored a technology, you could discuss future trends for the technology, or perhaps for the domain to which you have applied the technology. You could also outline ideas for future projects that would be a continuation of your own project.

Finally, some reflections on the project itself can also be included. During the project, you might have experienced that certain things did not work out as planned; maybe the technology did not live up to the expectations, maybe your choice of methodology turned out to be improper, or maybe there were other factors beyond your control that significantly impacted the project. Remember that it is much better to bring out any problems you have experienced into the open and discuss them, than it is to try to ignore or hide them. Being able to present and discuss problems makes it clear for the evaluators that you have recognised that the problem was indeed a problem; if you ignore it, the evaluators may doubt that you have even identified the issue as a problem… In short, a conclusion should be an honest retrospect on the project, with a clear tie-in to the original problem formulation.

What to avoid in a report

We have now discussed what to put into your report. There are definitely also things that you should avoid in your report. Obviously you should avoid bad spelling, bad grammar, etc., but there are also more subtle elements to avoid.

Probably the most important thing to avoid is **large sections of copied material**. Adding such sections without clearly indicating the source can be considered outright cheating, and you may simply fail the project for this reason alone! Note that this concept also covers "rewriting" of material. Just switching around some words in the copied material does not mean that it cannot be considered a copy. When you wish to draw in some external material, do it either by

- Adding a reference to the original material, and nothing more
- Adding a "summary" (in your own words) of the material, and still add the reference.

Note that this consideration also holds for figures, illustrations, code snippets, etc..

Also avoid **writing in bland, generic terms**. Do not introduce a concept in a sales-pitch manner like ***"Technology X enables you to rapidly develop streamlined and user-friendly websites with seamless integration to…."***. When introducting e.g. a technology, focus on concrete, factual properties that are relevant for the project at hand.

When drawing conclusions in a report, it is very easy to become **subjective**. If you are e.g. comparing two technologies for programming, and state that ***"I found technology X to be better than technology Y"***, then you need to back that claim with some facts. Was it better because it resulted in fewer lines of code, reduced the running time of the program, was faster to implement, etc. – what is your measure? Avoid subjectivity – rely on facts.

Many things can go wrong in a dissertation project, and some may be beyond your control, like losing data, sickness, etc.. It is okay to mention such things during the final reflection over the project, but avoid turning the report into a blame game, where you repeatedly refer to any problems you might have encountered, with statements like ***"If only this had not happened, I could have…"***, or ***"due to my PC crashing, I could not…"***. Harsh as it may sound, the evaluators will not give you a higher mark out of pity…

Planning

No two dissertation projects are the same, and the planning may be influenced by external factors, in particular if the project is done in cooperation with a company. Still, there are some general points that will apply to all projects.

The first point is simply to recognise that since the project is supposed to represent a work effort of 400 hours, and since the period available for performing the project is typically 10-11 weeks, it will require an effort comparable to having a full-time job in the entire period, even if the period e.g. extends across the Christmas holidays! That also means that having just a part-time job during that period will give you an extremely busy schedule. In the project period, the project must be your first priority!

A consequence of the above is also that there is no time for hesitation about getting started. If you hesitate and do not really get started before you have, say, 5-6 weeks left, it becomes increasingly difficult to put in the required work effort on 400 hours.

So, have a plan ready from day 1, and start working from day 1. One way of keeping the pace up is to make a plan with many small deadlines in it, and having the first deadlines after just a couple of weeks. Having concrete – and close – deadlines to work against is a useful way of keeping yourself going. You could e.g. have a deadline for having gathered materials, have a deadline for writing each section of the report, etc.. If your project is a "classic" development project, that development should also be divided into short iterations. Of course, you might have to change the plan as you go along; a plan should be an active tool through the entire project period, not something you just make the first day and then never look at again…

Finally, also plan to be finished with the actual work a few days before the actual deadline. You will need some time at the end to do reviews for structure, spelling, grammar, references, table of contents and so on (see the next chapter). And plan to be completely finished at the latest the day before the deadline, so you have room for encountering practical problems like a broken printer.

AND REMEMBER TO TAKE BACKUPS OF YOUR WORK!! Make a habit of doing that at the end of the day.

Finishing the project

Once the project deadline gets really close, you will – obviously – have to finish the project. That might sound trivial, but the point is rather: what should you spend your remaining time on during the last couple of days?

At this point in the project, many will probably be in the situation where they have not achieved all they set out to do. However, it is usually not a good idea to spend the last days frantically adding e.g. more functionality to a program or perhaps more pages to a website. The time is better spent on making sure that you end up with a proper report.

So what is meant by a "proper report"? Remember that the report is your tool for convincing the supervisor and the external censor that you have done a good job. It is wrong to reason e.g. that "if my program is good, it is less important that my report is also good". It is rather the other way around; if you have a poor report, it can be very hard to convince anybody that you have done a good job. So, even though it might seem like a detail, it is actually quite important that your report presents itself well.

First, the report should obey the structural requirements, as they were outlined in the chapter ***Report Structure***. The report should thus include:

- A **front page**, including: project title, name of group members, the name of the education, the name of the supervisor, and the project period
- An **abstract**, which is a short (at most half a page), self-contained summary of the project, including the main problem formulation, the main features of the methodology, and the main conclusions
- A **Table of Contents**, including page numbers
- An **introduction**, where you introduce the problem, the context in which the problem is solved, etc.
- A **problem formulation**, where you state explicitly what questions/problems it is that this project will specifically address.
- A **methodology**, where you formulate the details of the methodology you will use to address the questions/problems from the problem formulation
- (whatever sections you need for actually working on the problem – this depends on the nature of the project and the chosen methodology)
- A **conclusion**, where the main achievements and learning points are discussed in relation to the problem formulation
- A **bibliography**, with references to all the materials you have used. This can be books, websites, videos, etc..
- A **list of appendices**, with a short explanation of the purpose and content of each appendix
- The **appendices** themselves, if you have any

It is very hard to imagine a report where there should be any reason to not include any of the above, so use the above as a checklist before handing in the report.

Second, there are some less formal properties of the report that you should give attention to:

- **Spelling and grammar**: English is not the native tongue for almost all students at (international) CS and PBA WEB, so writing proper English can be quite a challenge. Of course, this education is not an education in written English, so no-one expects you to be able to write in perfect English. Still, if your English is so poor that it becomes hard to understand what you write, it will effect the evaluation negatively… Therefore, do put some effort into reviewing the language. Use whatever tools your software offers (Microsoft Word can do spelling and grammar checks, as can most other word processing tools), or maybe let somebody else read and comment the report.
- **Consistency**: During the project work, one can easily lose overview of the material. You might have found material, that you did not use anyway, but it still remains in the bibliography. You might have changed the numbering of figures, and have forgotten to change the references in the text, etc.. Such inconsistencies can be very confusing for a reader, so also spend some time for cheking and removing inconsistencies.
- **Visual structure**: Even if you follow the above report outline, you can still improve the visual appearance of the report by using proper page breaks (new chapters should start on a new page), avoid "orphan lines" (one or two lines sitting alone on a page), making sure your figures and illustrations are properly aligned and do not jump ahead of the text, etc.. . Even trivial things like using the same fonts and line spacings consistently improves the evaluators first impressions of the report. Again, having a pair of fresh eyes reading a draft print of your report is a great help. It does not need to be a person with any knowledge of the subject, since they should only focus on the presentation of the report.
- **Print quality**: The report has to be handed in as 3 printed copies. This is the version the evaluators (supervisor and censor) will see. So it is also important to ensure that the printed report itself has a reasonable quality. This is particularly important if your report contains many illustrations, which a poor printer may smother. If you have many illustration where color is important, you should of course make color prints of the report, even if this might be a bit more difficult and expensive. Finally, bind the report properly. Buy some good quality binders to put the report into – this is better than trying to staple the (probably rather thick) report together.

It might seem like a lot of additional work just for the sake of presentation, but again, remember that the report serves the purpose of convincing the evaluators that you have done a good job. Whether you like it or not, the presentation does matter… This is also why it is important to plan to stop the work as such some days ahead of the deadline, so you have time to properly finish the report as described above.

Getting a high mark

Figuring out the proper mark for a dissertation report is a delicate matter; in a ideal world, it should be a predictable and almost automatic process, but in practice it isn't. A dissertation report is not like a simple multiple-choice exam, where a computer can spit out a mark immediately after the exam. It is therefore also hard to define precisely what is takes to get a high mark. Still, there are some properties that a report must have in order to aspire to a high mark:

- **Consistency**: A dissertation report should represent one unified body of work, and it is therefore very unfortunate if inconsistencies can be found. Inconsistencies can be at multiple levels, from erroneous references to figures all the way up to describing a certain methodology and then in practice doing something entirely different.
- **Traceability**: it is not uncommon to see a project include some process element – like e.g. a SWOT analysis – which is carried out…and then not used for anything at all! If a certain process element is included, it should be possible to trace what the outcome of that process element was actually used for. Likewise, elements that are supposed to appear as the result of some sort of process (say, a web page design) should not just appear out of the blue.
- **Craftsmanship**: If you decide to include a certain process element, you should make sure that you fully understand its purpose, and how to use it. Using a process element in a wrong way is a clear indicator that you have not really understood it.
- **Clarity**: Even though a dissertation report is not an exercise in e.g. English grammar or style, clarity and correct use of language does improve the overall impression. An otherwise brilliant report written in a very convoluted way could end up with a low mark, simply because the message doesn't come across.

The above represents one way of looking at a report – another way to look at a report is in terms of the "level" of writing. In that context, we often talk about **the descriptive level, the analytical level, and the reflective level**.

Writing at the Descriptive level

At this level, you are only describing your actions, but you are not relating them to any kind of theoretical foundation. If you are developing an application, you might write something like *"these were the requirements for the program, so I developed this program, with the 17 classes in diagram xxx"*. Even though this might be a perfectly functioning application, the report still does not demonstrate that you have any sort of deep understanding of what you are doing. What kind of process were you following? What are your arguments for working in this particular way? And so on… Since the purpose of the dissertation work is to demonstrate that you can apply all the theories and methodologies you have learned to solve a practical problem – while doing this in a conscious manner and being able to argue for your decisions – a report written at a descriptive level will usually end up with a low mark. This surprises some students, who think that as long as the end goal was reached, the documentation of the "journey" towards it is less important.

Writing at the Analytical level

At this level, you can describe your actions and also relate them to a theoretical framework – that's more or less what analysis is defined as. If you are developing website designs, you have perhaps looked into some theories about usability, etc.., and figured out how you can apply those teories to your particular problem. Traceability is again of the essence here; if you discuss particular theories you wish to apply, you must also explicitly describe how you applied them. Some students think it is enough to refer to some theories more or less at random, and then just do what "seems proper", without relating back to the theory. That sort of writing does not lift your report to the analytical level. A solid report at the analytical level does on the other hand demonstrate that the student has a deeper understanding of the topics, and will often result in a mark in the mid-to-good range. Still, it takes a bit more to get the top marks.

Writing at the Reflective level

At this level, your depth of understanding is so great that you can actually start to critically discuss your own approach; you can reflect over your own work. This could e.g. mean that you decide to diverge from the process you have chosen, because you can provide and formulate good arguments for it (diverging without argumentation is definitely not reflection, rather sloppiness…). At the end of the project, you are also capable of doing a sort of retrospect of the project, where you can point out elements that could have been improved, and also ways of improving it. Being able to do these sorts of reflections can get your report up to the top end of the mark scale. Note, however, that this assumes that the report is already brilliant at the descriptive and analytical level. Just being able to state that *"I've done a terrible job, but at least I know it is a terrible job"*, does not propel your project into high mark territory.

So, you did everything you could, but you still did not get the mark you expected… Unfortunately, even if you do conform to the above recommendations, there is still no guarantees with regards to the mark. Remember that the final mark is decided by the supervisor AND an external evaluator (the "censor"), and that person may have different perceptions about what it takes to score a high mark. So even if you and your supervisor thinks that the project is almost perfect, the censor may not agree… There will therefore always be some level of subjectivity in the evaluation process, even if all parties involved try to avoid it.

The dissertation exam

The conclusion of the dissertation work is the dissertation examination.

When is the dissertation exam?

The actual date will of course vary from semester to semester. It is typically about two weeks after the deadline for handing in the project. The school administration will provide you with the specific date.

What is the format of the dissertation exam?

The exam is oral, and lasts up to a total of 30 minutes, including evaluation. The exam has two stages. The first stage is a presentation (see later) by you, where you have 10 minutes to talk freely and without interruption about the dissertation work. The second part - which will typically be about 15 minutes – is a "questions and answers" session, where the supervisor will ask you various questions about the project, for instance asking you to clarify unclear points in the report, put a broader perspective on certain topics, etc.. Note that in a group report, the group must do a joint presentation which lasts 10 minutes per student (e.g. 30 minutes for a three-student project). After the joint presentation, each student will then individually do a "questions and answers" session as described above.

Who is present at the exam?

The student(s), the supervisor and an external censor.

How is the final mark decided?

Before the exam starts, the supervisor and the censor evaluate the project, and assign a preliminary mark to the project. After this, the actual exam takes place:

The official descriptions of the educations describe the purpose of the oral dissertation exam as follows: ***the oral defence is primarily used to ensure that the deliveries are made by the student, and secondly for minor adjustments in the evaluation of the student's level***. In other words, the oral exam serves two purposes:

- To make sure that the student(s) actually did this project themselves
- To possibly adjust the mark given for the report before the exam

In practice, this means that the mark can only be marginally changed by the oral exam. So, a bad project cannot be saved by a good oral exam. Likewise, a good project cannot be "destroyed" by a less-than-perfect exam, unless the exam reveals some sort of cheating, like having copied large portions of the work from an external source.

How should I prepare for the exam?

As the exam contains the two stages described above, you should obviously both prepare a presentation (see below), and prepare for the "questions and answers" session. Preparing for the "questions and answers" session is best done by reviewing your own report with the intention of finding weak spots, since you will often get questions relating to such weak spots. A weak spot could be e.g. a claim that is not backed by any evidence or material, like claiming that ***"here we see that technology X is very easy to use"***, introducing a new concept without any explanation, drawing subjective conclusions, etc.. If you find such spots in your report, try asking yourself: why did I write this? If you can come up with an answer to that, you can probably also answer such a question at the exam. The supervisor might also ask you to put some perspective on certain topics, like if a technology could be used in a different context, if you could have used other tools for the job, etc.. Preparing for this will involve reading up on the used tools, methods, technologies, etc., and maybe see if you can find examples of how they have been used elsewhere.

What should the presentation contain?

The presentation – which is supposed to last about 10 minutes per student – will usually begin by giving an overview of the project. What was the main problem, how did we attack the problem, and what were the main conclusions. If you have an actual product like a program or website, a short demonstration is also fine. However, do not spend all your time on just this. Strictly speaking, all of the above should be found in the report as well, so you should also use the presentation to supplement the report with some more perspective, like if you could have done certain things differently, if you can see that some conclusions might be ambiguous or even wrong, and so on. In general, do not try to hide any problems you had during your project. It is much better to bring out the problems in the open and discuss them. By doing that, you demonstrate that you are aware of the problems, which in itself is positive. If you can even discuss ways of solving the problems, that is even better.

Finally, any generic advice on how to behave at an oral exam also applies here. No chewing gum, turn off your phone, dress seriously, act confident and polite, and so on. But at this point in your education, this should be old news.

The final mark

After the oral exam, you will receive one overall mark for your dissertation work.

The exam went great, but I still got a low mark… How is that possible?

Again, remember that one of the purposes of the exam is to make "***minor adjustments in the evaluation of the student's level***". In other words: a good exam cannot save a bad project. On the other hand; if the exam reveals any kind of serious cheating, like having copied the entire work, or significant sections of it, from an external source, you may fail the exam, even if the project is good. So even though the exam can only marginally change the final mark under normal circumstances (no cheating detected), the exam should not be taken lightly…

Why did the supervisor not warn me that I would get a low mark for my work?

The supervisor is formally not allowed to give you specific indications of the "level" of your work during the project period. The dissertation work is an exam, so the supervisor cannot give you any guarantees about a minimum mark before the actual exam. The supervisor will do his/her best to guide and advice you and give you feedback during the project, but that does not translate to any guarantees about the final mark. Asking the supervisor questions like ***"Can I be sure to get at least 10 for this project?"*** is meaningless, and the supervisor will not answer them…

I am so unsatisfied with my mark that I want to complain!

If you are really unsatisfied with your mark, there is the possibility to file a complaint. However, doing that is a two-way bet; you might actually end up with a lower mark than what you originally got. So before filing a complaint, do a bit of honest self-reflection, and see if you still really think that the mark was unfair. Did I really put in the required effort? Did I actually do all the things required for a good project? Did I do the work myself, or did I copy some of it? In practice, we see very, very few actual complaints about the dissertation marks.