Pedagogical Portfolio — Robert Feldt

Robert Feldt

August 6, 2012

Abstract

This document describes the pedagogical experience of Robert Feldt. Together with a selection of relevant example documents and certificates attached to this document it constitutes Robert’s pedagogical portfolio.

1 Teaching in undergraduate and graduate courses

I have taught a total of 18 different courses (at a total of 33 different course occasions) at three different Swedish universities; Chalmers University of Technology, University West and Blekinge Institute of Technology. Table 1 gives key information about them while table 2 (at the end of this document) describes my level of involvement in different course activities. The percentages in table 2 are relative to the total workload for each activity. Note that course creation is not always applicable; I only list the percentages of course creation for the 14 course occasions I have been involved in when the courses have been newly developed.

I taught at Chalmers University of Technology during my PhD studies in 1997–2002, as a guest teacher in 2008–2009 and then as an associate professor (senior lecturer) from August 2009 onwards. I have been involved in a total of 8 different courses (C1–C9 in tables) at a total of 16 different course occasions. For 8 of these course occasions the course in question was newly developed or revised as to be a new course. As can be seen my involvement has steadily grown and I have recently (re-)created three new courses at Chalmers. I typically also do all lectures and act as examiner in the recent courses; the rest of the lecturing is typically guest lecturers I invite from my industrial research contacts.

I was an assistant professor and senior lecturer in Software Engineering at Blekinge Institute of Technology from December 2006 and onwards. I have been involved in a total of 4 different courses (B1–B4) at a total of 6 different course occasions.

I was an assistant professor and senior lecturer in Software Engineering at University West, Sweden from October 2003 to November 2006. I was involved in a total of 5 different courses (H1–H5) at a total of 11 different course occasions.

The course C6 was chosen by Chalmers from a list of courses I proposed to them in 2008 when they had asked me to be a guest teacher. It was chosen to best complement the existing courses given at the Master Programme in Software Engineering. I created the course together with my colleague Tony Gorschek and we shared giving the lectures. I was sole examiner and course responsible. Three students took the course as a PhD course with higher requirements.

The course C8, on Requirements Engineering, was created from scratch for the new, joint master programme in Software Engineering at Chalmers and Gothenburg University. The course has been very well received with a record 180+ students enlisted its inaugural year. For the 2012 occasion of the course, it is being re-designed to make extensive use of short, instructional videos to be studied
by students outside of lecture and workshop time. This will enable a more real-time, problem-solving and discussion-oriented format for the full-class sessions. This format has been tried in the C8 and C9 courses in 2010-2011 with very effective results and is now taken to the next level. The C8 course has also been given in parallel to PhD students by being adapted to their specific research projects.

The courses C4 and C5 were initiated by and partly created by me. For the course C5 I directly contacted Watts Humphrey, the SEI creator of the PSP and TSP courses, and got the TSP material as the first non-US university.

For the course B4 I developed a lecture and a task on doing research paper reviews. I gave support for PhD students in conducting the task and evaluated their results. The course C7 had a similar structure but I developed the material for all seminars and evaluated all the students’ results.

All courses except C7, B4 and B3 have been ‘full’ university courses corresponding to 7.5 ECTS credits. B3 is the Master Thesis in Software Engineering course that accompanies the students final-semester, master thesis projects and typically span 6-10 months of study for the average students, corresponding to 30 ECTS credits. However, I acted as the single examiner of all thesis and had introductory as well as supporting lectures during their projects. B4 and C7 was shorter, PhD level courses corresponding to 3 ECTS credits but given over longer time stretches than master level courses.

The format I have evolved over the years for my master and bachelor level courses are typically that I reduce the number of traditional lectures and give more weight to a practical project as well as workshops that prepare for the project. The project is typically carried out in a group of 4-8 students depending on the number of available resources. I also complement with 1-2 practical tasks that students do individually. The projects involve two components in how they are examined, one focus on the actual result and application of knowledge from the course on the practical task while the other requires the students to step back and reflect on what worked well, less well and why. I often use both a text book early in the course and then gradually base the course on more recent

<table>
<thead>
<tr>
<th>Id</th>
<th>Course</th>
<th>Students</th>
<th>Level</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>C9</td>
<td>EDA397, Agile Software Processes</td>
<td>75</td>
<td>Master</td>
<td>2</td>
</tr>
<tr>
<td>C8</td>
<td>DAT165, Requirements Engineering</td>
<td>70-180</td>
<td>Master &amp; PhD</td>
<td>2</td>
</tr>
<tr>
<td>C7</td>
<td>Scientific Publication in SE</td>
<td>3</td>
<td>PhD</td>
<td>1</td>
</tr>
<tr>
<td>C6</td>
<td>DAT165, Software Product Line Eng.</td>
<td>30</td>
<td>Master &amp; PhD</td>
<td>2</td>
</tr>
<tr>
<td>C5</td>
<td>EDA380, Team Software Process</td>
<td>45</td>
<td>Master</td>
<td>1</td>
</tr>
<tr>
<td>C4</td>
<td>EDA395, Personal Software Process</td>
<td>20–50</td>
<td>Master</td>
<td>3</td>
</tr>
<tr>
<td>C3</td>
<td>EDA375, Modern Software Engineering</td>
<td>45</td>
<td>Master</td>
<td>1</td>
</tr>
<tr>
<td>C2</td>
<td>TIN420, Software Engineering</td>
<td>50</td>
<td>Master</td>
<td>2</td>
</tr>
<tr>
<td>C1</td>
<td>EDA221, Real-time Systems</td>
<td>100</td>
<td>Bachelor</td>
<td>2</td>
</tr>
<tr>
<td>B4</td>
<td>Research, Publication and Review</td>
<td>20</td>
<td>PhD</td>
<td>1</td>
</tr>
<tr>
<td>B3</td>
<td>PA2403, Master Thesis in Software Eng.</td>
<td>40–50</td>
<td>Master</td>
<td>3</td>
</tr>
<tr>
<td>B2</td>
<td>PAD013, Research Methodologies in CS. &amp; SE.</td>
<td>160</td>
<td>Master</td>
<td>1</td>
</tr>
<tr>
<td>B1</td>
<td>PAD009, Master Thesis in Software Eng.</td>
<td>50</td>
<td>Master</td>
<td>1</td>
</tr>
<tr>
<td>H5</td>
<td>OSB050, OO SW Development with UML</td>
<td>50</td>
<td>Bachelor</td>
<td>1</td>
</tr>
<tr>
<td>H4</td>
<td>PMC020, Software Dev. through Modeling</td>
<td>45</td>
<td>Bachelor</td>
<td>2</td>
</tr>
<tr>
<td>H3</td>
<td>PPC010, Managing SW Engineering Projects</td>
<td>45</td>
<td>Bachelor</td>
<td>2</td>
</tr>
<tr>
<td>H2</td>
<td>KVD010, Req. Eng., Verif. &amp; Valid.</td>
<td>60</td>
<td>Master</td>
<td>3</td>
</tr>
<tr>
<td>H1</td>
<td>PVD010, Software Engineering</td>
<td>60</td>
<td>Master</td>
<td>3</td>
</tr>
</tbody>
</table>
research results that students study directly from research papers. Together these course elements strikes a balance between available resources (some traditional lectures early in the course are ‘cost-effective’ and help give overview and orientation to the area), being realistic/’job-like’ (projects is how software engineers typically work) while giving theoretical background (textbooks to give background and overview early and then advanced and recent concepts from research papers) and deepened understanding (reflection on project work as well as discussion of the different projects and outcome at the end of the course).

I want students to be mentally active and engaged in my lectures, rather than copying or simply writing down notes, and I want them to be able to catch up or revisit material after lectures and workshops. Thus I very frequently record lectures and workshops and upload videos with screen captures of slides and pictures of whiteboards that have been used. Students frequently give feedback that this helped them go back and really go deep on issues they might have missed during lectures and discussions. I have seen a clear quality increase in students results in the last years as they have also videos from earlier years courses to study in the ongoing course. For my next set of courses I will build on this further by making the basic course material more easily available in video format and then using even more of time in class and workshops for discussion and collective problem solving.

Summary: I have extensive experience in all parts of course teaching (initiating and creating new courses, developing new pedagogical formats, lecturing, instruction and in supporting lab and project work) at different university levels (bachelor, master and PhD). I have a broad knowledge base and can teach in a majority of subjects within the Software Engineering and Computer Science area. I can quickly develop new courses and teach them in many different forms.

2 Supervision of theses and research

2.1 Supervision of PhD students

I am / have been involved in the supervision of a total of ten PhD students (3 finished PhDs, 2 finished licentiate). I currently supervise 6 PhD students. Tables that show the details about these students can be found in my CV.

Richard Torkar was affiliated with University West (HV) in Trollhttan, Wasif Afzal, Shahid Mujtaba, Michael Unterkalmsteiner and Bogdan Marculescu was/are affiliated with Blekinge Institute of Technology (BTH) while the rest are affiliated with Chalmers University of Technology (CTH).

I am/was formally the main supervisor for Shahid Mujtaba, Ali Shahrokni, Emil Brjesson, Bogdan Marculescu and Hajar Kashfi. For Ana Magazinius and Richard Torkar I acted as main supervisor in practice. Ana Magazinius finished her thesis project in August 2011 and was on parental leave until 120301 after which she printed and defended her thesis. Shahid Mujtaba works at Volvo and will finish his lic in parallel to his full time day job.

For Richard Torkar, Ana Magazinius and Joachim Pernsthl I was asked to become supervisor after they had finished their licentiate theses. The reason was that they all had lacked proper, software engineering specific supervision. My supervision was instrumental in making Richard and Ana finish their studies successfully. I attach letters from Richard Torkar and Sofia Börjesson (Ana’s examiner) to support this claim.

Even when I’m not the main supervisor I have tended to have a very active supervision role and have actively participated in both conducting studies, writing and commenting on papers as well as in the strategic decisions around the students.

Richard Torkar has stayed in academia after his PhD and is now an associate professor at University of Gothenburg. Wasif Afzal has stayed in academia after his PhD, first as a post doc at Blekinge Institute of Technology and now as an assistant professor in his home country of Pakistan.
My style of supervision is to be very supportive and involved and focus also on the individual and not only at the tasks at hand. I want to inspire and motivate my students. If I can manage that they can often tap into their unknown/unused potential and outperform themselves. I try to be very hands-on and act as a guide in the early stages of their project and at the start of each individual study but then gradually be less involved and let them build confidence in their own abilities.

I put a lot of effort into establishing realistic views and expectations from the start. I continuously update and refine my ‘Guide to Starting a Phd’ which is freely available online:

http://www.cse.chalmers.se/~feldt/advice/feldt_guide_to_starting_a_phd.pdf

and is used not only by my own PhD students but also by Master level students and by PhD students from other universities (I know this through email feedback). I also frequently ‘micro-blog’ about new pedagogical ideas in relation to Software Engineering on my Twitter feed as found on my home page.

2.2 Supervision of Master and Bachelor Theses

I have supervised a total of 35 Master Theses in Software Engineering; they are listed in an attached document. I currently supervise another three Master theses that are expected to finish in 2012. I have been the examiner for more than 60 other Master Theses at BTH.

While at HV I also supervised 5 Bachelor Theses in Software Engineering, two of which won the yearly ‘Best Bachelor Thesis Award’. The bachelor theses are listed in Appendix A.

3 Theoretical knowledge, Pedagogical studies and development projects

I have done several pedagogical studies and development projects in three different main contexts: Diploma of Higher Education at Chalmers, an NSHU-sponsored pedagogical project called PIFF, and when developing new courses. The latter is described in Section 1 above and in Section 8 below, the other two contexts are detailed in the following subsections.

3.1 Diploma of Higher Education, Chalmers

During 2003 and 2004 I took courses at Chalmers towards a diploma of higher education (‘docentfräberedande kurs’). Below I briefly describe what I learnt from each course. Because of increasing teaching duties during 2005 I never got to finish my Diploma at Chalmers. In January 2009 I submitted this portfolio and some of the papers I developed as part of the PIFF project for the final 2 points needed for the diploma.

3.1.1 TLC102 Teaching and Learning in Higher Education

This course is the equivalent of a two-week full-time or two point (old system) graduate course. It concerns the planning, presenting and evaluation of courses. It focuses on the importance of student learning and ways of aligning course objectives with teaching methods and assessment of learning.

From this course I developed a much deeper knowledge of the connections between what we want students to learn and how we assess that learning. By making the assessments ‘authentic’, i.e. more realistic and in line with actually using the knowledge in practice and ‘in the field’, we can ensure that students learn even if they only try to pass the exam. Ideally, the assessment should be formative rather than summative. A formative assessment is a help in the students learning and not simply a check that the learning has been achieved. This has affected me to add more realistic projects to my courses.
For me it was also important that the course introduced the Bloom taxonomy of educational objectives [3]. I have found this taxonomy to be an important way of thinking about different aspects of learning when planning and developing a course. I have used the updated version of the Bloom taxonomy when developing rubrics for evaluation of Master thesis projects [2]. It can be used in most course content discussions and planning to evaluate what kind of learning and knowledge we want and currently are helping students to do.

The TLC102 course also opened my eyes to Higher Education as a scientific discipline in itself. This started me on a mission to try to keep up-to-date on new developments not only in my own field but also on related results in the field of higher education.

### 3.1.2 TLE202 Supervision of Research: Principles, Models and Issues

This course is the equivalent of a two-week full-time or two point (old system) graduate course. It concerns supervisory contracts, giving and receiving feedback, managing conflict, the different phases in the PhD process, quality assurance, gender equity, multiculturalism and research ethics.

This course opened my eyes to the fact that there are many aspects in research supervision that do not directly have to do with the research task at hand. Even though success in supervision can follow from an almost ‘blind’ focus on the research task, this can be a big risk when facing challenges along the way. By being aware of and taking other issues into account one can create a more stable supervising relationship.

The course also taught me the importance of trying to clarify expectations up front, both as a supervisor and as a student. If this is not done, it is very likely that there are very different views on what has been agreed upon during the project. This can lead to disappointment and lack of trust or energy. By discussing expectations up front some of this risk can be reduced. This has for example helped shape the knowledge on expectation discussion and definition in master thesis project that we have captured in our support framework (see Section 3.2).

My interview with an experienced supervisor, John Hughes at Chalmers, was interesting. John does not have a conscious ‘model’ for how to supervise, he works side-by-side with the student actually doing research together. Thus he does not think one should take on more than two to three students at a time. The level of commitment he showed for his students was remarkable. Even so he allowed the students much freedom and discussed at length that the initiative to the research should come from them.

Based on my interview and the course I developed a hypothesis that the ‘freedom-foremost’ approach to supervision gives a larger variability in the outcome. When a student succeeds in this model, the result can be of higher quality since it is more a product of the student’s own, less paradigm-polluted ideas than of the supervisor’s. The flip side is that not every student will step up to this task with the additional freedom. It requires more initiative and drive from them. On the other hand, a supervision style that is more managed by the supervisor may reduce the variability in quality of the outcome and thus ensure that a larger percentage of students really make it. However, there is a risk that the better students do not realize their maximum potential as easily in this style.

In this course I also got to explore one of the problems I have had in supervision: that of flooding students with too many ideas. I developed a new view in which the supervisor should act more as a guru for the student. The focus should be to together build something (the PhD) that can build good things (research), rather than only to build a good thing (the thesis).

### 3.1.3 TLC101 Pedagogical Project

This course is the equivalent of a four week full-time or four point (old system) graduate course. In the course the participants studies and discusses the differences between quantitative and qualitative
research models, and plans, formulates, completes, reports and reflects on a pedagogical project within their own teaching.

My project focused on how to get students to learn continuously throughout a course instead of cramming just before the exam. I re-created a traditional Software Engineering course to have nine small tasks over the 7 weeks of the course. All work counted towards the grades. The evaluation showed that a majority of students thought it was helpful to their learning. It also encouraged them to think and question the material more.

My reflection on the course is that it is much fun and very rewarding as a teacher to turn your pedagogical development into research projects! Not only can you write up and present pedagogical projects like any other research, you also understand more of what is happening, and improve more, by using a scientific method. By being systematic, measuring outcomes and writing down results and conclusions, you get more out of the teaching. This has had a profound impact on the way I view both my research and teaching, and lead to many later pedagogical projects and papers.

3.2 PIFF project for improving SE Master Theses, BTH

Together with Gordana Dodig-Crnkovic and Frank Lüders at Mälardalen University (MDH) and Martin Höst at Lund University (LU) I currently run a pedagogical development project. The project is funded with 2 million Swedish krona from NSHU (Swedish Agency for Networks and Cooperation in Higher Education) and with 0.5 million Swedish krona from the three participating universities (BTH, LU and MDH). The project started in September 2007 and ran to September 2009. The projects name was PIFF (‘Programvaruexjobb för Industri- och ForskningsFramgång’, in English ‘Software Theses for Industry and Research Success’).

Diploma degrees are awarded by academia, while diploma work is often done in collaboration with industry or a research group. The aim of the PIFF project was to improve knowledge exchange between academia, industry and research during diploma work, supporting both a student and an advisor in the different phases (planning, execution and grading/assessment) of diploma work in Software Engineering. The results of the project will be generalizable to other multi- and cross-disciplinary fields.

The following reports and publications was written/published during the PIFF project:


Although my work with improving the Master Thesis course at BTH had already started prior to the PIFF project the project helped in refining the course in how to better adapt to industrial master
thesis projects and what is valued by the companies. This knowledge has later been important at Chalmers were a majority of projects are done in collaboration with industry.

For my pedagogical evolution the PIFF project was important since it showed me that there is immense value in actually doing research and publishing also my pedagogical activities. In particular

4 Teaching activity outside the university

I frequently teach Software Engineering-related subjects to engineers and personnel in industrial companies. This is a challenge since the audience often have very different backgrounds and have little time. Often they are not so interested in basic knowledge but want to better understand the state of the art in some SE sub-area.

The key solution I have found in these situations is to adapt the material to the practical experience of the audience. Preferably you should collect specific examples of problems, processes or methods that they use daily, and connect your material to that. As an example, at a recent talk given at RUAG (formerly SAAB) Aerospace Sweden AB in Gothenburg, I introduced recent research results on software inspection techniques by contrasting it with the type of inspections they use. This was very effective and they claimed they got several ideas on how to optimize their inspection methods, from my presentation.

In later years I work more directly in helping the companies actually implement new knowledge in their development processes. I have used a combination of developing and demonstrating tools, to giving advice and support and feedback on their own education and improvement efforts.

A selection of teaching and presentations that I have taken part in outside of the university:

- Spring 2012, Ericsson AB, Karlskrona: ‘Optimizing System Testing’
- 2010-02-12, SAAB Security AB, Gothenburg: ‘What we know about Agile and Test-Driven Development’
- 2008-12-14, RUAG (formerly SAAB) Aerospace Sweden AB, Gothenburg: ‘Software Engineering and Verification and Validation Research Trends’
- 2008-11-20, Swedish Association on Software Testing Q4 meeting: ‘Connections between Requirements and Testing’
- 2008-03-11, IBC Euroforums course ‘Future of Test Management’: ‘Upgrading your testing game’
- 2007-10-17, Expo-C Roadshow for Developers, Testers and Managers: ‘Upgrading your game - state-of-the-art software research and why you should care?’

IBC Euroforum did a questionnaire-based evaluation after their course. On a scale from 1-5 (where 1 is bad, 4 is very good and 5 is excellent) I got a score of 4.12 for ‘Delivery’ and 4.08 for ‘Content’. 
5 Assessment of teaching contributions from the student perspective

In course evaluations students often characterize me as ‘inspiring’, ‘thought-provoking’ and ‘stimulating’. I consider these my main educational strengths together with my ability to get a holistic view of a subject and present a coherent framework for the knowledge involved. A weakness is that I sometimes forget what the more difficult parts of a subject are and go too fast so that students sometimes find it hard to follow. In course evaluations at HV I generally got high scores with 4.1 on average (on a 5-point scale where 5 is best/highest) on ‘pedagogical ability’.

An example of a representative course evaluation from a course at Chalmers is attached to this portfolio. Despite the course and written exam being rated as very hard and demanding by students, I got a 4.41 rate on ‘Pedagogical skills’ and 4.44 on ‘Ability to answer questions’. I was the examiner and co-creator of the course. The course as a whole was rated 3.83 on ‘Overall quality’.

The newly developed Requirements Engineering course (C8) at Chalmers in 2011 got a course evaluation score of 4.07 (on a five point scale with 5 corresponding to ‘Excellent’). According to the program responsible at the time this was the highest overall score for any course given at the Software Engineering program. Individual teachers in courses are no longer individually evaluated in course evaluations at Chalmers but since I developed the course and gave it solely with assistance from one of my PhD students, and we were both positively mentioned in course evaluation comments, I think this speaks to my pedagogical abilities.

5.1 Course improvements based on student feedback

The most common complaints in my courses is that the workload is ‘too high’, that the ‘project group did not function well’ and that ‘feedback took too long to get back’. I know that I’m quite demanding as a teacher and that my courses require a lot of effort from students. However, with high expectations and good support I see students again and again that outperform even what they themselves would have thought possible. However, I try to improve on this aspect by better syncing with the teachers in other courses that students take at the same time. By trying to reduce the amount of deadline overlap between courses the stress level of students can often be reduced.

Malfunctioning groups is very hard to avoid. Students often have very varying background and does not always find suitable working formats. I have tried in recent years to use individual surveys and personality tests to gauge the background, experience and ‘style’ of students before dividing them into groups but it is still too early to evaluate if this has really made a difference. In the mean time I try to support malfunctioning groups and guide and help them overcome difficulties so that it does not negatively affect their learning. However, I also try to explain to them that this is very common also in their future work situations; often engineers are assigned to projects based more on actual competence rather than to make the most optimal group. This is a reality we have to help prepare students for rather than try to avoid.

When it comes to giving feedback this has been a challenge in recent years when the ‘models’ used to assigning teacher hours have lead to less total hours available per course. I try to overcome this by using more peer assessment in my courses. For example, in the last Requirement Engineering course students first wrote requirements in assignment 1 and were then randomly assigned to review the requirements of other students in assignment 2. Since both writing and reviewing of requirements is an important part of Software Engineering the assignments both had pedagogical value while reducing the amount of time the teachers had to spend on reviews and corrections. I will develop similar solutions for my other courses in the future.

I have also addressed the feedback issue partly by using Twitter extensively to give up-to-date information in my courses. Examples can be found on the course pages with URL’s listed below.

1On a five-point scale where 1 is ‘Very bad’, 2 is ‘Bad’, 3 is ‘Ok’, 4 is ‘Good’, and 5 is ‘Very good’
6 Production of teaching materials

I have produced many different types of teaching materials. In the following I briefly touch on just a small selection of them. More is available upon request.

While supervising master theses at HV and BTH I collected different tips and tricks on how to succeed. Since many students had similar problems I collected them into a single document. I have attached the tips and tricks document to this portfolio.

Example of my videos and other course material used in my recent Chalmers courses can be found on their respective home pages:

http://www.cse.chalmers.se/feldt/courses/reqeng/

and

http://www.cse.chalmers.se/feldt/courses/agile/

7 Administration and pedagogical leadership

I have over two years of experience as the head of the Master programme in Software Engineering at University West, Sweden. During this time, I re-created the curricula of the programme, to have a broader set of courses and better covering all aspects of the IEEE SWEBOK [1]. I also added more project-based elements to the courses and made these projects more like real-world software development exercises. A big change for the programme during my years as the head was an adaptation to international students.

8 Reflections on pedagogical activity

My view on education is that it should be clear, hands-on, based on real-world problems and full of real-world examples. It should use discussion as a means of summarizing and reflecting on the acquired knowledge. The role of the teacher is not to push out knowledge (in)to the students but to give them a framework and context for pulling/acquiring the knowledge. Thus, to help them understand and broaden their internal models by exemplifying the knowledge. The teacher should also lead discussions about issues raised and conclusions drawn. In this type of teaching, the teacher is more of a mentor and guide than an ‘instructor’ in the traditional sense. This view has permeated my teaching experience over the years.

During my years at Chalmers University of Technology (CTH) we transformed the existing SE course into a more practical, project-based course. We also introduced a PSP (Personal Software Process) and a TSP (Team Software Process) course. They all used actual software development in student projects as the basis for discussing the course material. For the PSP and TSP courses I directly contacted Watts Humphrey at the Software Engineering Institute at Carnegie Mellon University. Watts is the creator of the PSP and TSP processes and books. As one of the first universities we got a draft of Humphreys TSP book and developed our course based on that.

At University West (formerly University Trollhttan/Uddevalla) I continued along these lines when working with the courses on SE, Object-oriented Analysis/Design, UML, Requirements Engineering and Verication and Validation and in thesis supervision. When I have created and/or recreated courses it has been to introduce a project-based format where the lectures, seminars and laborations support the learning that goes on within the project.

At Blekinge Institute of Technology I have focused on learning outcomes and how to define them. This is challenging, especially in Master Thesis projects which is the crowning achievement of a long education. As such it should show that the student have not only gained the subject
knowledge she needs, but also that she has acquired generic skills, such as presentation, time management, writing etc. I have used rubrics with clearly defined quality criteria and levels to support both students and supervisors in better understanding thesis quality.

My educational ability has been further advanced by the three courses I have taken for Prof. Michael Christie at CTH towards a Diploma in Higher Education (‘docentfrberedande kurser’). The main changes I have made in my teaching after taking these courses is to plan for and support continuous learning even in theoretical courses by having many small preparatory/study tasks and by being very explicit with what kind of knowledge, skills and attitudes the students should have acquired after the course. The prep tasks can be either practical (try a method/technique in practice) or reflective (think about the consequences of some theoretical material). In my pedagogical project for Michael I showed that both of these types are important and that students think the preparatory/study tasks throughout the course helps them learn better than the traditional form with an exam in the end.

One of the courses within the Diploma program was on supervision of research and PhD students. This was an important course in helping me get a broader view of the issues that can arise in research supervision. I’ve had lots of use of this knowledge in my supervision of Bachelor, Master and PhD students.

During 2005 I was selected as one of Sweden’s top ten ‘Promising university teacher’s/researcher’s’ for the Council of Higher Educations (Rådet för Högere Utbildning, RHU) Summer school. My project during the summer school was on how we can make the generally useful abilities/skills of university education more explicit and thus more directly supporting their continued development and evolution in students during their university programmes. I continued this work in my position as Head for the Master programme in SE by extending the courses with elements that supports the students in developing their presentations and creative thinking skills. The work on generic skills has continued within the PIFF pedagogical project, with a focus on defining and supporting students in developing such skills for and in their Master theses. The more general idea of defining learning outcomes has come in handy in the transition that many European Universities have done / are doing within the Bologna process.

For the future I would like to work more on making even better use of the traditional lecture format. I think traditional slides are not very powerful learning (and thus teaching) aids. The main problem is that teacher tends to use it to write down text instead of using more visualization and diagrams to help support the ‘text’ that is spoken by the teacher. I want to find interactive computer visualizations for the key concepts of a course and implement them within visualization software. I want to evaluate how such a format compares to traditional slides. Longer-term, and given that the visualizations are successful, I plan to work this into a kind of book (but possibly only available online or in downloadable formats) called ‘Software Engineering Visualized. In part I use my current course improvements that use video material in different form as a learning step towards this more ambitious, visualisation- and explanation-driven approach. However, over time video is not interactive enough as can be seen in the recent rise and popularity of internet-based university courses given by e.g. Udacity and Coursera. To further hone my skills in this type of pedagogical formats I have been following Udacity courses during the spring of 2012 and will take a Coursera course during the autumn of 2012. I can see a benefit in their approaches but also a niche since they mainly focus on basic, bachelor-level courses. There is a clear potential for me (and Chalmers!) to use such pedagogical methods also for more advanced courses close to research.

In the future I would also like to work more on formative assessment and how to structure courses so they can be more agile and adapt to students needs/complaints/ideas during the course. At BTH I lead a student who developed an online course feedback and evaluation system that can be used from mobile phones. The summary statistics from student evaluations is available right away from a web page, also accessible from a mobile phone. I would like to explore and evaluate how
this can be used to support student learning.

9 Subject area knowledge

I have a broad knowledge of SE and related areas and can create and teach courses on almost any type of subject, from technical and low-level things up to ‘softer’ issues related to psychology or organization. I started programming computers at a very early age and sold my first software solutions already in my teens. I have then worked as a software consultant in my own business since 20 years. Combined with my theoretical studies and research work I can confidently say I’m at or very near the state-of-the-art in a large number of software engineering areas. Here is a list of my personal and professional interests in relation to software engineering:

- **Software (Reliability) Engineering**: Software development methods, (Automated) Software Testing, Software and network fault tolerance, Software and design diversity, Redundancy, Robustness, Limits of machine reliability, Program Compilation and Optimization, Parsing and program transformation, Software processes, Formal methods, Software Verification and Validation, Software Management, Human-Computer interaction, Software Visualization, Parallel software, Fault-tolerance in parallel systems, Web programming

- **Practical psychology of teams, design and invention**: Teamwork and team leading, Mental processes of invention and design, Innovation, Creativity and flow, Philosophy of software, Software psychology, Open-Source development processes

- **Artificial Intelligence, Optimization & Exploration**: Evolutionary, Genetic and Memetic algorithms, Machine Learning, Ensemble systems, Neural networks, Symbolic/Logic Reasoning Algorithms

- **Mathematics**: Statistics, Statistical modeling, Bayesian probability theory and statistics, Design of experiments, Information theory, Complex and adaptive systems

- **Biological system**: Evolutionary theory, Population genetics, Diversity, Analogies between biology and computers, Evolutionary computation, Genetic Programming, Bioinformatics

10 Other pedagogical qualifications

I have attached a recommendation letter from Prof. Michael Christie and the diploma from the RHU Summer School to show some external views on me as a teacher and educator.

A Supervised bachelor theses

- Daniel Persson & Kjell Petterson, ‘En prototyp för ett presentationsverktyg med programmerbara animationer’, HV 2006
- Andreas Höglund, ‘Om möjligheten att automatiskt generera tester likt en människa’, HV 2005, Best bachelor thesis award
- Olle Johansson, ‘Plugin-Based Automated Testing Tool for .NET Assemblies’, HV 2004
- Sanny Jacobsson, ‘Autonomic Computing: Dependable solution for increased IT complexity?’, 2004
B  Documents attached to this portfolio

The following documents are attached to this portfolio:

- Course evaluation and summary for DAT165, Chalmers, 2009-01-08
- Recommendation letter from Michael Christie
- Diploma from RHU’s Swedish Summer Institute for the Improvement of Teaching and Learning 2005 ‘Learners for Change’
- Certificates for my involvement in some of the PhD student’s supervision:
  - Richard Torkar
  - Sofia Börjesson (for student Ana Magazinius)
  - Johan Karlsson (for student Ali Shahrokni)
- Robert Feldt, ‘Some advice for a successful thesis project’, version 1.3, 2006-12-21
- List of Master Theses Supervision

References


<table>
<thead>
<tr>
<th>Id</th>
<th>Occasion</th>
<th>Role</th>
<th>Created</th>
<th>Lectures</th>
<th>Instr.</th>
<th>Lab/Proj</th>
</tr>
</thead>
<tbody>
<tr>
<td>C9</td>
<td>2012</td>
<td>Examiner</td>
<td>80%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C8</td>
<td>2011</td>
<td>Examiner</td>
<td>90%</td>
<td></td>
<td>25%</td>
<td></td>
</tr>
<tr>
<td>C9</td>
<td>2011</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td>25%</td>
<td></td>
</tr>
<tr>
<td>C8</td>
<td>2010</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td>25%</td>
<td></td>
</tr>
<tr>
<td>C7</td>
<td>2009</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>C6</td>
<td>2009</td>
<td>Examiner</td>
<td></td>
<td></td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>C6</td>
<td>2008</td>
<td>Examiner</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>2001</td>
<td>Course Responsible</td>
<td>50%</td>
<td>50%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td>1999</td>
<td>Assistant</td>
<td>75%</td>
<td>50%</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>1999</td>
<td>Course Responsible</td>
<td></td>
<td></td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>1998</td>
<td>Assistant</td>
<td>50%</td>
<td>20%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>1998</td>
<td>Assistant</td>
<td>30%</td>
<td>10%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>1998</td>
<td>Assistant</td>
<td>50%</td>
<td>50%</td>
<td></td>
<td>50%</td>
</tr>
<tr>
<td>C2</td>
<td>1997</td>
<td>Assistant</td>
<td></td>
<td></td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>C1</td>
<td>1997</td>
<td>Assistant</td>
<td></td>
<td></td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>C1</td>
<td>1996</td>
<td>Assistant</td>
<td></td>
<td></td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>B4</td>
<td>2008</td>
<td>Lecturer</td>
<td>25%</td>
<td>25%</td>
<td></td>
<td>25%</td>
</tr>
<tr>
<td>B3</td>
<td>ht08</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3</td>
<td>vt08</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3</td>
<td>ht07</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td>2007</td>
<td>Examiner</td>
<td>50%</td>
<td>60%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>vt07</td>
<td>Examiner</td>
<td>50%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>H2</td>
<td>2006</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td>H5</td>
<td>2005</td>
<td>Examiner</td>
<td>50%</td>
<td>50%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H4</td>
<td>2005</td>
<td>Examiner</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H3</td>
<td>2005</td>
<td>Examiner</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2</td>
<td>2005</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>H1</td>
<td>2005</td>
<td>Examiner</td>
<td>100%</td>
<td></td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>H4</td>
<td>2004</td>
<td>Examiner</td>
<td>100%</td>
<td></td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>H3</td>
<td>2004</td>
<td>Examiner</td>
<td>100%</td>
<td></td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>H2</td>
<td>2004</td>
<td>Examiner</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>H1</td>
<td>2004</td>
<td>Examiner</td>
<td>100%</td>
<td></td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>H1</td>
<td>2003</td>
<td>Examiner</td>
<td>100%</td>
<td></td>
<td></td>
<td>100%</td>
</tr>
</tbody>
</table>
Course Evaluation & Summary – SPLE (DAT165), LP2 HT2008

This is the course evaluation summary for the DAT165 course, Software Product Line Engineering (SPLE), given within the Master Programme of Software Engineering at Chalmers in LP2 of HT2008. The course had two major parts, a set of lectures and a larger project where students did an SPLE process assessment at an industrial software company. The project was supported with several “support times”/lectures where one of the teachers answered questions about the project and lectured/detailed how the students could solve specific issues. The support times were also used to show and discuss trends of SPLE research and state-of-the-art.

Summary of course and proposed changes

The course was judged a success with students giving it a 3.83 grade for “Overall quality” even though the course was considered hard. The lecturers were given grades of ~4.5. The course was considered to be hard and 46% of students failed the written exam. Several commented that they were not used to actually working with “real problems in industry”; this was seen as very rewarding and important but also made the course harder.

Students that gave the course an overall quality score of 3 (there were no lower grades) commented that “project assignment was too big”, “it is not good that the course continues into January”, “too big assignment”, “not more than 2h lectures per day”, “course is big” and “course was abstract”. Other comments on the course were that “the contact to a company is gold, not only for this subject but for self-experience, thanks for that!”, “Very good course, one that you are proud to have chosen”, “Good layout overall, good teachers, good course”.

Student numbers were somewhat hampered by the fact that the course was announced very late and did not get in to the normal course management system at Chalmers.

Based on the course, its results and the course evaluation we propose the following changes for next year:

• Announce the course earlier and to everyone in the target audience
• Add a lecture and a lab session where students uses a commercial SPLE tool (like Gears or similar)
• Add or retarget a lecture to prepare students better for how software companies are organized, how industry differs from academia etc
• Try to have only one lecture per day
• Consider having separate, but shorter, support sessions for each group

It is not possible to keep the project assignment in industry and finish it before Christmas; the extra setup times in working with a company and their tight schedules, makes this impossible. The benefits of the assignment are so big though that we strongly urge next years’ teachers to keep it.
For the teachers responsible for the Software Engineering educational program we urge them to consider how to better bridge between academia and the software-intensive industry in more courses; it is both needed and appreciated.

**Basic information about the course**

<table>
<thead>
<tr>
<th>Course title:</th>
<th>Software Product Line Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course code:</td>
<td>DAT165</td>
</tr>
<tr>
<td>Period:</td>
<td>LP2, HT2008, Chalmers</td>
</tr>
<tr>
<td>Examiner:</td>
<td>Robert Feldt</td>
</tr>
<tr>
<td>Lecturers:</td>
<td>Robert Feldt, Tony Gorschek</td>
</tr>
<tr>
<td>Assistant:</td>
<td>Martin Ivarsson</td>
</tr>
<tr>
<td>Students signing up:</td>
<td>~35 students signed up for course, ~8 of them dropped out after the first lectures (citing “high workload this period”)</td>
</tr>
<tr>
<td>Students taking the written exam 15/12:</td>
<td>24</td>
</tr>
<tr>
<td>Students finishing the project on time:</td>
<td>22</td>
</tr>
<tr>
<td>Students that took the course evaluation:</td>
<td>18</td>
</tr>
</tbody>
</table>

**Course evaluation**

A course evaluation sheet was handed out after students had finished their written exam. It had 2 questions for each of the involved teachers (“Pedagogical Skills” and “Ability to answer questions”), four questions for the course and its contents (“Lectures”, “Support times”, “Exam”, “Overall course quality”), and one open-ended comment question. The table below summarizes the answers to the numerical questions. Comments from the open-ended question were given in the summary above.

<table>
<thead>
<tr>
<th>Question</th>
<th>Average</th>
<th>Answer alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF – Pedagogical skills</td>
<td>4.41</td>
<td>1-Very bad, 2-Bad, 3-Ok, 4-Good, 5-Very good</td>
</tr>
<tr>
<td>RF – Ability to answer qs.</td>
<td>4.44</td>
<td>As above</td>
</tr>
<tr>
<td>TG – Pedagogical skills</td>
<td>4.67</td>
<td>As above</td>
</tr>
<tr>
<td>TG – Ability to answer qs.</td>
<td>4.56</td>
<td>As above</td>
</tr>
<tr>
<td>MI – Pedagogical skills</td>
<td>2.94</td>
<td>As above</td>
</tr>
<tr>
<td>MI – Ability to answer qs.</td>
<td>2.59</td>
<td>As above</td>
</tr>
<tr>
<td>Course – Lectures</td>
<td>4.00</td>
<td>As above</td>
</tr>
<tr>
<td>Course – Support times</td>
<td>3.06</td>
<td>As above</td>
</tr>
<tr>
<td>Course – Exam</td>
<td>2.11</td>
<td>1-Very hard, 2-Hard, 3-Ok, 4-Easy, 5-Very Easy</td>
</tr>
<tr>
<td>Course – Overall quality</td>
<td>3.83</td>
<td>1-Very bad, 2-Bad, 3-Ok, 4-Good, 5-Very good</td>
</tr>
</tbody>
</table>
9 November 2004

To whom it may concern

This is to say that Robert Feldt has been known to me for over a year and has participated in pedagogical courses here at Chalmers. I know he has extensive teaching experience and would benefit greatly from the Summer Institute run by the Council of Renewal for Higher Education.

Robert is an enthusiastic teacher with many innovative ideas and would be an asset to the group work and the group dynamic at the Summer Institute. By accepting him I think that RHU would also be providing inservice experience to a staff member from a regional university and this I think would benefit both RHU and HTU’s work and profile.

Robert has a very pleasant personality and enquiring mind and these qualities would also be an advantage in terms of his engagement with the work of the Institute.

Yours faithfully

Michael F Christie

Docent, CKK
The Swedish Summer Institute for the Improvement of Teaching and Learning

June 5\textsuperscript{th} to 10\textsuperscript{th}, 2005
‘Learners for Change’

Organised on behalf of the Council for the Renewal of Higher Education.

Robert Feldt participated in the Swedish Summer Institute 2005 with the theme ‘Learners for Change’. The participants are a selected group of young academics chosen for their outstanding research, strong commitment to teaching, and the promise of a successful academic career. Participants are selected from among all Swedish institutions of higher education and they come from a cross section of the various academic disciplines. The Summer Institute was designed especially to support the participants’ interest in the field of teaching and learning.

Major themes addressed during the Summer Institute 2005 included: Assessment in higher education; Perspectives on learning in higher education; Curriculum design; Strategies for change both in terms of the higher education context and in terms of ‘learners for change’. More importantly, the participants also worked with a project, based on a problem from their own teaching they brought to the week. This process, which drew explicitly on the rich diversity of participants’ experiences, facilitated the sharing of different perspectives and the articulation of the relevance of the themes of the week. The participants are supposed to continue working with this project at their universities and departments during the autumn.

In his participation in the Summer Institute, Robert Feldt has shown a strong commitment to the development of teaching and learning in higher education.

Sunne, June 10\textsuperscript{th}, 2005

Magnus Gustafsson, PhD
National Project Manager
Chalmers University of Technology

Lars Haikola, Professor
Chairman
The Council for the Renewal of Higher Education
Certificate Regarding Dr. Robert Feldt’s Supervision of Richard Torkar

Dr. Robert Feldt supervised Richard Torkar from early 2004 to June 2006 at which time Richard defended his PhD thesis at Blekinge Institute of Technology.

Even though Robert was formally assigned as supervisor, he acted as main supervisor in practice and he took far greater responsibility than was actually expected from his side. In the end, Richard could finish his thesis well ahead of time due, to a large extent, to Robert’s supervision.

Karlskrona, January 24, 2012.

Dr. Richard Torkar

Associate professor (docent)
Blekinge Institute of Technology
S-371 79 Karlskrona
Sweden
Certificate – Ph D Candidate Supervision

Robert Feldt has since 2009-12-01 together with myself been supervisor for Ana Magazinius. We have worked jointly with her, both taking the roles as main supervisors. A year after the Licentiate degree, the team around Ana changed and Robert entered in December 2009 as supervisor with expert knowledge within her subject area (software engineering and in particular cost estimation). The supervision was since then intense from both Robert and myself and we shared the main supervision responsibility although Robert in addition was also responsible for the direct content-wise supervision.

In view of above, it is without doubts that Robert Feldt has acted as main supervisor during the period 2009-12-01 to Ana Magazinius’ defense 2012-04-10.

Göteborg, 2012-05-05

Sofia Börjesson, Professor and examiner.

Sofia Börjesson, Professor in Technology Management (Chair)
Research Director Center for Business Innovation
Technology Management & Economics
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
+46 31 772 51 59
+46 70 760 7673
sofa.borjesson@chalmers.se
http://www.cbi.chalmers.se
http://www.chalmers.se/bme/EN/organization/divisions/Innovation-engineering
To whom it may concern,

Concerning Robert Feldt’s role in the supervision of Ali Sharokni from 2008 to present date

Robert Feldt has been involved in the supervision of Ali Sharokni since June 2008. When Ali started his PhD studies in June 2007, he was supervised by Peter Öhman and myself. Robert was appointed assistant supervisor to Ali after Peter had left the department in early 2008.

Robert is now the main supervisor to Ali, while I serve as Ali’s examiner. Although it took a while before Robert was officially appointed main supervisor, it is my duty and pleasure to certify that he in all respects, and with great skill, in practice has served as the main supervisor to Ali since June 2008.

Sincerely,

Johan Karlsson
Some advice for a successful thesis project

Robert Feldt, Version 1.3, 2006-12-21

In this document you find some advice, ideas and recommendations to increase your chance of a successful thesis project. These guidelines combine experience gained from writing, supervising and examining several B.Sc. and M.Sc. projects at Blekinge Institute of Technology, University West and Chalmers University of Technology in Göteborg. The guidelines elaborate on the most common mistakes made by B.Sc. and M.Sc. candidates. It is expected that students who are yet to start their thesis projects will be able to avoid many of these mistakes by following the advice in this document. Note that this document should be read in addition to the course book; the course book contains important information not found here so you need to study them both!

Write early, write often, write all the time

The central result from your thesis project is the knowledge you gain and the knowledge you can convey to others. Your main way to do the latter is with the thesis itself; the written account on what you did, why you did it, how you did it, what results it gave and which significance it has in a bigger context.

If you start writing your thesis the last couple of weeks before the final report is due, the result will not be very good. The main reasons are that you will not remember the details of what you did and especially not why you did it. Another reason is that good writing requires good thinking; it's a process of continuous refinement that you cannot rush in a short amount of time. Therefore you must start writing directly when your project starts and continue writing throughout your project. Furthermore, the earlier you start writing, the earlier you have something to show your advisor so he/she can give relevant feedback.

Focus, focus, focus

It is ok to attack a large problem if you focus your efforts on a small part. If you try to do too much you will not have the time to finish anything (or start anything!) and thus the outcome will not give any new insight or knowledge. Better to focus your efforts on one part and then take it from there; maybe it will even be easier than you thought so that you can expand your scope as you go along.
A clear indication that you are trying to do too much is that you do not know where to start; there are simply too many things to be done for you to choose. Step back, decide on a number of steps to take and then just do it.

**Write a log**

During the course of a (technical) project there is a multitude of information that you need to collect. You search research databases and find interesting articles, you read an article and jot down some notes, you get ideas for the design of your solution or questions you need to seek answers to, and you make decisions that affects the final system/outcome/result. A good way to collect all this information is by writing a technical log/diary of your progress.

Writing a log ties in nicely with the advice to “Write early, write often, write all the time”. The log can be your main vehicle for writing. From the log you get the “seeds” to go into your thesis. You copy an URL or article reference into the reference list, you copy your notes on a paper into the “Related work” section where you summarize the findings in the paper and how it relates to your work, and you copy design ideas and expand on them to describe your system etc.

The log and the thesis complement each other. While the former documents your progress and thinking with the details of the why and the what, the latter is the polished, refined and distilled final form of your ideas and results. Together they encompass the range of thinking and writing activities you need to go through to succeed. It is important to remember though that the final report can never be in the form of a log/diary.

**Write with the reader in mind**

The main problem with technical reporting is lack of clarity. The purpose of a report is to present information and knowledge for someone else. It is your responsibility to present that information as clearly as you possibly can; you should do everything in your power to make things easy for the reader.

Readers are busy. While you write a single report, there are many different things and texts that attract the attention of a reader. Easing the task of the reader should be your main concern.

With readers we do not only mean the examiner and your adviser (although easing their task is of course your main concern ;)); No one should start a thesis project with the goal of only writing for your examiner. Why limit yourself like that? Your goal should be to do a project interesting enough to have a large group of potential readers. So you need to ease the reading for all of them. If your report or thinking is unclear you will have problems attracting broader interest even if your work is of good quality.

**Think logically and clarify your line of thought**

Clear writing requires an ability to think logically. You should be able to write down (or map out in a visual form) the ideas, decisions and arguments that constitutes your “line of thought”. The line of thought is your essential argument. It should be rational and you should be able to motivate each step of it. If you make it very clear for yourself it will be so much easier to convey it to others.

**Be honest and modest**

Apart from the moral and ethical problems with hiding or concealing information or right out misrepresenting facts there is a more pragmatic reason to be honest: readers will be grateful that you point out the problems with your work. By giving a frank and open statement of the weak sides you make it easier to assess the generality of the results. Why would you want to claim your work is stronger than it really is? If you mislead a reader they will eventually find out the problems anyway (and be the more angry with you for making them waste their time). Over hyping will just lead to frustration. Be honest and modest.

**Be bold**

Even if you should be modest when you report on your results you should be bold when choosing and defining your project. You know more and can do more than you think! Don't just look at some gurus (be they world-known researchers or a student in your class) and amaze yourself with the quality of their work; focus on your tasks and work hard to do a good job with each one of them. You will probably be amazed with what you can achieve.
Consider alternatives

There is never a single right way to do something. There are always many alternative ways. Some of them are somewhat better or somewhat worse than the others but there is seldom a single one that is “the best”. Readers will think of alternatives when they read your paper and will want to know why you chose to do what you did and the trade-offs it entailed. You should try to anticipate as many alternatives as you can and make sure that you understand your own choices. Not only will it be easier for you to defend your main line of thought later, it will also make it easier to choose a good one from the outset.

About writing style

The preferred writing is in the passive voice. That is, you do not refer to yourself or to your group. Use formal language. Informal style is not acceptable. The following example is a bad example of writing:

“I decided to study firewalls because security is cool. Therefore I looked at some books on firewalls, read a bit of www.firewalls.org, and configured a couple of MS Windows firewalls in my home computer. It was interesting”.

This is sloppy, hard to read and puts the focus on you and your feelings and not on what you have done. Instead this text would be better if written as:

“Due to wide-spread use of computer networks computer security aspects play an important role in the modern society. Firewalls are one of the mechanisms to achieve secure communication and protect network services. The goal of this project is to study the most common firewall techniques and compare their performance. The study is based on several literature and web sources, as well as on a comparison of some firewall installations. The results of the project are the following: … ”

Have a logical structure

Naturally, the report must contain all the necessary parts, such as abstract, introduction, methodology, results, discussion and conclusions. An example of each of those parts can be found in any published scientific article and in courses related to scientific writing.

The report must have a logical structure. This means, that if you e.g. introduce concepts A and B, and concept B is a derivative of concept A, then you must first introduce concept A and then concept B.

This structure should also be present at lower levels. Within one section each paragraph follows from the previous one. Within one paragraph each sentence follows from the previous one etc.

Use proper references and describe important concepts

Whenever you introduce something new, for example, a fact, a term, or a technique, you either describe it yourself or provide a complete reference to a document that contains a comprehensive and authoritative description of it. Usually, well-known broad terms like deadlock, waterfall process model and middleware do not require references, but a description is necessary. All the important concepts that are crucial to understand your project needs to be defined in the report. The report should be self-coherent, i.e. a reader should not be required to look up other sources to understand it.

Use a spellchecker

Do not hand in any writing with minor errors that disturbs the reader. The reader will not be able to follow your line of thought and give feedback on the bigger picture since the minor errors will distract. Spelling errors is a typical example that is easily fixed. There are spell checkers in all modern document editors. Be sure to use it.

Avoid plagiarism

In an undergraduate project it is likely that much of what you do in your project have been previously covered by others. This is an opportunity for you to strengthen your argument, justify your choices and show how your work ties in with the current state-of-the-art. You achieve all this by providing proper and timely references. Occasionally, you can also use a quotation from a text, but only if you explicitly mark it as such and state from which source it is taken. However, quotations are an exception and should be used sparingly.

Whatever you do, you should never present other peoples ideas, designs, texts, paragraphs, sentences, tables, diagrams, source code, systems, solutions, results and so on without referencing them. If you make someone
else’s work look like your own it is plagiarism and, when detected, will be punished. Copying from others, without giving them proper credit, is considered a very serious academic offence. If you do it, you risk a lot. Not only will you fail the thesis course, you will not be able to finish your degree.

**Intermediate versions of your report**

It is absolutely necessary to present written results before every meeting with the advisor. It is not necessary to write an exhaustive report on everything you show at a meeting, especially in the beginning of the project. However, whatever you show must be well motivated and explained. The intermediate versions of your report will grow into your final report so the writing you do for them directly helps to produce a final report of high quality.

Remember, no matter how good you are your skills are worthless if you cannot explain what you are doing, how and why. Written reports are necessary for you and for the advisor/examiner. Your advisor has other students to take care of and the reports are necessary to make a record of your progress. You benefit from the intermediate reports by identifying potential problems in early stages of the project and by preparing yourself for the final report.

**When in doubt or stuck, ask**

Don’t be stalled for many days because you do not know how to proceed. Think about the problem and clearly map it out so you understand what the problem is. Do your “homework”; try to find information that might help you solve the problem. Use the Internet and the library. But don’t be afraid to ask. Your adviser is there to help you. He/she will help you but it is hard for him/her to understand that you are “stuck” and need help. While it is his/her responsibility to help you it is your responsibility to ask. Do that.

**Weekly status emails**

To help you work continuously it is a good idea to have at least a weekly contact with your advisor. This can be in the form of an email answering the following questions:
- What have you done this week?
- How many words your report currently contain? New and changed since last week?
- What plan to do next week?
- Which, if any, problems you are currently struggling with? What do you need help with?

Your email can be the basis for a short discussion with your advisor and help him/her help you. However, please note that it is up to you and your advisor to decide how and what kind of meetings and contacts you will have. You should decide on this and write it into the thesis proposal.

**Additional information**

Some references related to the advice in this text:

R. Feldt's Master Thesis Supervision

I have extensive experience in supervising (38 total out of which 35 to completion and 3 ongoing) and examining (75+) Master Theses in Software Engineering and Development. The projects I supervise frequently lead to published research papers at major conferences and journals. This can be beneficial both to your own learning and career, the company you work with and the university. It is also frequently seen as positive when you apply for jobs. This is in addition to the quality benefits for your thesis project; aiming for publication will often help you focus your efforts and increase the quality of your work.

Master Theses - Currently Supervising


MT-37. Behnoush Pejhanmanesh and Narjis Hachilif, "Visualization of Software Quality Trend", Chalmers 2012 (with Ericsson AB, Gothenburg)


Master Theses - Supervised

MT-35. Viktor Fritzon and Tobias Alette, "Effects of code metric visualization and information radiator", Chalmers 2012 (with RUAG Space AB)

MT-34. Godwin Sebabi Semwezi, "On the automation of negative testing", Chalmers 2012 (with Ericsson AB, Kista)


MT-29. Mehwish Rashid, "Evaluating the Effectiveness of Regression Test Selection Techniques in Industry", Chalmers 2011 (with Ericsson AB in Karlskrona)


MT-25. B. Marculescu, "Implementing a Software Verification and Validation Management Framework in the Space Industry", Chalmers 2010 (with RUAG Space AB)


MT-23. M. Zulfijar & A. Naseer, "Investigating Exploratory Testing in Industrial Practice - A Case Study", BTH 2010 (with Sogeti AB)


MT-15. D. Larsson, "Challenges and Solutions in Test Staff Relocations within a Software Consultancy Company", BTH 2007 (with Softhouse AB)


MT-12. O. ur Rehman Malik, "Mining Object-Oriented Software Execution Traces for Patterns for Automated Testing", HV 2006


MT-7. A. Alexandersson, "RubyComp - a Ruby-to-LLVM Compiler Prototype", HV 2004

MT-6. J.-Å. Hedström, "RubySharp - a Ruby to CIL Compiler", HV 2004

MT-5. K. Lyngfelt, "MorphR - a Morphic GUI in Ruby", HV 2004

MT-4. P. Olofsson, "CheckR - Extendible Static Analysis Tool for Ruby", HV 2004

MT-2. B. Maroszek, "Contour - a Ruby-to-Flash bridge for presentations", HV 2004