Pedagogical Portfolio — Robert Feldt

Robert Feldt

December, 2015

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

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 in Software Engineering, Computer Science and Psychology and can teach in a majority of subjects within the Software Engineering and Computer Science area with short notice. I can quickly develop new courses and teach them in many different forms. More details to support this is given in the following.

I have taught a total of 18 different courses (at a total of 34 different course occasions) at three different Swedish universities; Chalmers University of Technology (CTH), University West (HV) and Blekinge Institute of Technology (BTH). 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, then docent, then professor) 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 reasearch contacts.

I was an assistant professor, then senior lecturer and then associate professor (docent) 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

Table 1: Courses taught

Id	Course	Students/year	Level	Times
C9	EDA397, Agile Software Processes	75	Master	2
C8	DAT165, Requirements Engineering	65-180	Master & PhD	3
C7	Scientific Publication in SE	3	PhD	1
C6	DAT165, Software Product Line Eng.	30	Master & PhD	2
C5	EDA380, Team Software Process	45	Master	1
C4	EDA395, Personal Software Process	20–50	Master	3
C3	EDA375, Modern Software Engineering	45	Master	1
C2	TIN420, Software Engineering	50	Master	2
C1	EDA221, Real-time Systems	100	Bachelor	2
B4	Research, Publication and Review	20	PhD	1
B3	PA2403, Master Thesis in Software Eng.	40–50	Master	3
B2	PAD013, Research Methodologies in CS. & SE.	160	Master	1
B1	PAD009, Master Thesis in Software Eng.	50	Master	1
H5	OSB050, OO SW Development with UML	50	Bachelor	1
H4	PMC020, Software Dev. through Modeling	45	Bachelor	2
H3	PPC010, Managing SW Engineering Projects	45	Bachelor	2
H2	KVD010, Req. Eng., Verif. & Valid.	60	Master	3
H1	PVD010, Software Engineering	60	Master	3

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 has 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 strecthes 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 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.

2 Supervision of theses and research

2.1 Supervision of PhD students

I am / have been involved in the supervision of a total of 11 PhD students (6 finished PhDs, 3 finished licentiate). I currently supervise 3 PhD students but 2 of them are planned to finish during 2016. 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, Hajar Kashfi and Per Lenberg. 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.

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, for Joachim it was a collaboration with Tony Gorschek.

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 Professor at University of Gothenburg. Wasif Afzal has stayed in academia after his PhD, first as a post doc at Blekinge Institute of Technology, then as an assistant professor in his home country of Pakistan and now as an assistant professor at Malardalens Hogskola in Vasteras. Ana Magazinius works at a research

institute; the rest of my PhD students now have roles in industry. Emil Alegroth and Michael Unterkalmsteiner are doing post-docs at Chalmers and Blekinge Inst of Technology, respectively.

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.robertfeldt.net/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 38 Master Theses in Software Engineering; they are listed in an attached document. I currently supervise another Master thesis that is expected to finish in 2015. I have been the examiner for more than 60+ other Master Theses at BTH and another 40+ at CTH and HV.

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. Recently I was awarded a 2 Million SEK (MSEK) project to develop new pedagogical methods to teach industrial engineers to become intrapreneurs and innovators inside of their companies based on the latest research knowlege. The project is funded by Vinnova and Chalmers but Chalmers has opened up for including BTH if I become appointed at BTH.

3.1 Diploma of Higher Education, Chalmers

During 2003 and 2004 I took courses at Chalmers towards a diploma of higher education ('docent-frberedande 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:

- M. Höst, R. Feldt and F. Lüders, "Support for Different Stakeholders in Software Engineering Master Thesis Projects", IEEE Transactions on Education, vol. 52, num. 2, pp. 288-296, 2010, doi:10.1109/TE.2009.2016106.
- G. Dodig-Crnkovic, F. Lüders, M. Höst and R. Feldt, "Improved Support for Masters Thesis Projects in Software Engineering". Rapporter frn NSHU, 2010.
- R. Feldt, M. Höst and F. Lüders. 'Generic Skills in Software Engineering Master Thesis Projects: Towards Rubric-Based Evaluation', In 22nd IEEE-CS Conference on Software Engineering Education and Training (CSEE&T'09), February 17-19, 2009, Hyderabad, India.
- R. Feldt, M. Höst, F. Lüders. 'Generic Skills in Master Thesis in Software Engineering', in Swedish Conference on Generic Skills in Higher Education, August 2008.

- R. Feldt. 'Improving a Master Thesis course for Bologna and Industrial and Academic Relevance', NU 2008 Education for a new time, May 2008.

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 and summer 2012, Ericsson AB, Karlskrona: Three seminars on 'Optimizing System Testing'
- 2011-04-27, Spotfire/TIBCO, Gothenburg: 'Agile Development and Automated Testing - State-of-the-art and Ways Forward'
- 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'
- 2007-10-23, Lindholmen Software Development Day, Gothenburg: 'Search-based Software Testing'
- 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'[1]. 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.

During 2012 I gave the 'Agile Development Processes' course at Chalmers. It got an overall score of 4.1 with positive comments like 'By far Robert Feldt is the most valuable teacher in this Master programme. In addition to that, the guys (Ali and Emil) were quite well. Thank you Robert and keep up with your good work', 'Preserve to next year: Available video and sound from lectures. The debate workshop', and 'Robert's team is perfect'. It was especially encouraging with the comment about the debate workshop which this was a new format I created especially for that course.

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

[1] On a five-point scale where 1 is 'Very bad', 2 is 'Bad', 3 is 'Ok', 4 is 'Good', and 5 is 'Very good'

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.

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 preporatory/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 preporatory/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. Ive 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 Högre 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 reasearch.

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 software technology and engineering 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 more than 25 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 technology and 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
- Maria Samuelsson, 'Attityder om programvaruutveckling en förstudie i hur programvaruutvecklares inställning påverkar deras arbete', HV 2004, **Best bachelor thesis award**
- 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'
- Diploma of Higher Education, Chalmers, 2009

References

[1] Alain Abran, James W. Moore, Pierre Bourque, and Robert Dupuis. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. IEEE Computer Society, Los Alamitos, CA, USA, 2004.

[2] L. W. Anderson and David R. Krathwohl et al. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon, USA, 2000.

[3] B. S. Bloom. *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. David McKay Co. Inc., New York, USA, 1956.

Table 2: Involvement in different course occasions/activities ('Instr.' = non-lecture instructions/workshops)

Id	Occasion	Role	Created	Lectures	Instr.	Lab/Proj.
C9	2012	Examiner		80%		
C8	2011	Examiner		90%	25%	
C9	2011	Examiner	100%	100%	25%	
C8	2010	Examiner	100%	100%	25%	
C7	2009	Examiner	100%	100%		100%
C6	2009	Examiner				50%
C6	2008	Examiner	50%	50%		50%
C4	2001	Course Responsible		50%		50%
C5	1999	Assistant	75%	50%		50%
C4	1999	Course Responsible		50%		
C4	1998	Assistant	50%	20%		
C3	1998	Assistant	30%	10%		
C2	1998	Assistant	50%	50%		50%
C2	1997	Assistant			50%	50%
C1	1997	Assistant			50%	50%
C1	1996	Assistant				50%
B4	2008	Lecturer	25%	25%		25%
B3	ht08	Examiner		100%	100%	
B3	vt08	Examiner		100%	100%	
B3	ht07	Examiner		100%	100%	
B2	2007	Examiner	50%	60%		
B1	vt07	Examiner	50%	100%	100%	
H2	2006	Examiner		100%	100%	50%
H5	2005	Examiner		50%	50%	
H4	2005	Examiner	100%	100%		
H3	2005	Examiner		100%		
H2	2005	Examiner		100%	100%	100%
H1	2005	Examiner		100%		100%
H4	2004	Examiner		100%	100%	
H3	2004	Examiner	100%	100%		
H2	2004	Examiner	100%	100%	100%	100%
H1	2004	Examiner		100%		100%
H1	2003	Examiner		100%		100%