TKK Research Reports in Computer Science and Engineering A TKK-CSE-A4/09
Espoo 2009

CHALLENGES OF TEACHING AND STUDYING PROGRAMMING AT A UNIVERSITY OF TECHNOLOGY — VIEWPOINTS OF STUDENTS, TEACHERS AND THE UNIVERSITY

Doctoral Dissertation

Päivi Kinnunen

Dissertation for the degree of Doctor of Philosophy to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium E at Helsinki University of Technology (Espoo, Finland) on the 11th of December, 2009, at 12 noon.

Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Computer Science and Engineering

Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan laitos

Distribution:
Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Computer Science and Engineering
P.O. Box 5400
FI-02015 TKK
FINLAND
URL: http://www.cse.tkk.fi/
Tel. +358 9 451 23228
e-mail: pakinnun@cs.hut.fi

ISBN 978-952-248-194-8
ISBN 978-952-248-195-5 (PDF)
ISSN 1797-6928
ISSN 1797-6936 (PDF)
URL: http://lib.tkk.fi/Diss/2009/isbn9789522481955/

Multiprint Oy
Espoo 2009

ABSTRACT OF DOCTORAL DISSERTATION	HELSINKI UNIVERSITY OF TECHNOLOGY P.O. BOX 1000, FI-02015 TKK http://www.tkk.fi
Author Päivi Kinnunen	
Name of the dissertation Challenges of teaching and studying programming at a university of technology – Viewpoints of students, teachers and the university	
Manuscript submitted June 2, 2009	Manuscript revised October 21, 2009
Date of the defence December 11, 2009	
☒ Monograph	☐ Article dissertation (summary + original articles)
Faculty Faculty of Information and Natural Sciences Department Department of Computer Science and Engineering Field of research Computing education research Opponent Professor Mordechai Ben-Ari Supervisor Professor Lauri Malmi Instructor Professor Emeritus Veijo Meisalo	

Abstract

This thesis analyses the challenges of the instructional process at a university of technology from the viewpoints of students, teachers and the university administration. The first research question concerns the difficulties students encounter when they study computer programming. Special attention was given to the students' reasons for dropping out of the introductory programming course (CS1). The second research question concerns computer science teachers' conceptions of studying and teaching. The third research question concerns how the instructional process was seen at the teaching organisation level. These three viewpoints represent holistic approach to the challenges of the instructional process.

General System Theory (GST) was used as the framework throughout the theoretical and empirical parts of this thesis. Three analysis models were developed: the "*dimension doughnut"*, the *three-layered didactic triangle* and the *feedback loop.* These tools were used as starting points for developing the categorisation of earlier literature based on its didactic focus, and to analyse systematically the collected empirical data. The empirical data was collected from students in the introductory programming course, computer science teachers, representatives of the administration and formal documents. Both qualitative and quantitative research methods were utilised.

The results suggest that the students' reasons for dropping out of the CS1 course were manifold and that they tend to cumulate. On average, dropped out students reported ten reasons that contributed to their decision, of which four affected their decision critically. The reasons included: course arrangements, difficulties to understand course topics, difficulties with time management, no consequences for dropping out, and preference for other courses. Computer science teachers' conceptions of studying were often content oriented. The teachers considered theory and concepts of computer science and the ability to apply knowledge to be the most difficult for the students. The experiences from the previous years courses and customs affected greatly the teaching process. Due to the large-scale courses, it was difficult to make adjustments to the ongoing course. The analysis of the formal documents and interview data of the administrative personnel revealed problematic aspects. For example, goal setting and planning were substance oriented. As a result, soft skills were not systematically discussed or taught during the studies. Moreover, the collection of feedback was not systematic and collected feedback was not always utilised. The results highlighted how students', teachers' and organisation's instructional processes interrelated.

Keywords teaching, studying, learning programming, introductory programming, drop-out reasons, instructional process, teaching organisation	
ISBN (printed) 978-952-248-194-8	ISSN (printed) 1797-6928
ISBN (pdf) 978-952-248-195-5	ISSN (pdf) 1797-6936
Language English	Number of pages 259 p.
Publisher Department of Computer Science and Engineering	
Print distribution Department of Computer Science and Engineering	
☒ The dissertation can be read at http://lib.tkk.fi/Diss/2009/isbn9789522481955/	

VÄITÖSKIRJAN TIIVISTELMÄ

TEKNILLINEN KORKEAKOULU
PL 1000, 02015 TKK
http://www.tkk.fi

Tekijä Päivi Kinnunen

Väitöskirjan nimi Opetus- ja opiskeluprosessin haasteet ohjelmoinnin opiskelussa teknillisessä yliopistossa – opiskelijoiden, opettajien ja yliopiston näkemys haasteista

Käsikirjoituksen päivämäärä 2.6.2009

Korjatun käsikirjoituksen päivämäärä 21.10.2009

Väitöstilaisuuden ajankohta 11.12.2009

☒ Monografia

☐ Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)

Tiedekunta Informaatio- ja luonnontieteiden tiedekunta
Laitos Tietotekniikan laitos
Tutkimusala Tietotekniikan opetustutkimus
Vastaväittäjä professori Mordechai Ben-Ari
Työn valvoja professori Lauri Malmi
Työn ohjaaja professori emeritus Veijo Meisalo

Tiivistelmä

Tutkimus analysoi opetus-opiskeluprosessin haasteita Teknillisessä korkeakoulussa opiskelijoiden, opettajien ja korkeakoulun näkökulmasta. Ensimmäinen tutkimuskysymys käsitteli opiskelijoiden kokemia vaikeuksia ja keskeyttämissyitä ohjelmoinnin peruskurssilla. Toinen tutkimuskysymys käsitteli tietotekniikan opettajien käsityksiä opiskelusta ja opettamisesta. Kolmas tutkimuskysymys kohdistui korkeakouluorganisaation näkemykseen opetus-opiskeluprosessista. Nämä kolme näkökulmaa edustavat kokonaisvaltaista lähestymistapaa opetus-opiskeluprosessissa esiintyviin haasteisiin.

Teoreettisena viitekehyksenä käytettiin yleistä systeemiteoriaa (GST). Tutkimusprosessin tuloksena on kolme analyysimallia: "dimensio donitsi", kolmikerroksinen didaktinen kolmio ja palautesilmukka. Näitä malleja käytettiin esimerkiksi olemassa olevan tutkimuksen kategorisointiin ja opetus-opiskeluprosessia koskevan empiirisen aineiston analysointiin. Työn empiirinen aineisto kerättiin ohjelmointikurssin opiskelijoilta, tietotekniikan opettajilta, opetushallinnossa työskenteleviltä henkilöiltä ja virallisista dokumenteista. Työssä käytettiin sekä kvalitatiivisia että kvantitatiivisia tutkimusmenetelmiä.

Opiskelijoilla oli useita samanaikaisia ja kumuloituvia syitä keskeyttää ohjelmointikurssi. Opiskelijat raportoivat keskimäärin kymmenen keskeyttämissyytä, joista neljä vaikuttivat keskeyttämispäätökseen ratkaisevasti. Syitä olivat: kurssijärjestelyt, vaikeus ymmärtää kurssilla käsiteltyjä asioita, vaikeudet ajanhallinnassa, ei seurauksia keskeyttämisestä ja muut kurssit. Opettajien käsitykset opiskelusta painottuivat asiasisältöön. Opettajien mukaan tietotekniikan käsitteet ja kyky soveltaa tietoa olivat opiskelijoille vaikeita. Opettajien opetusprosessiin vaikuttivat edellisiltä vuosilta kerätyt kokemukset sekä yliopistossa vallitsevat käytännöt. Kursseilla, joilla oli useita satoja opiskelijoita, oli vaikea tehdä korjauksia opetusprosessiin kurssin aikana. Opetushallinnossa työskentelevien henkilöiden haastattelut toivat esiin useita ongelmallisia asioita, esimerkiksi tavoitteiden asettelu oli sisältöpainotteista ja sen vuoksi muita tavoitteita (kuten ryhmätyötaidot) ei opetettu systemaattisesti. Yliopisto-organisaation palautteen keruu ei ollut johdonmukaista eri lähteistä eikä palautetta käytetty aina hyväksi. Aineiston analyysi toi myös esille miten opiskelijoiden, opettajien ja yliopisto-organisaation opetus- ja opiskeluprosessit olivat yhteydessä toisiinsa.

Asiasanat opetus, opiskelu, ohjelmoinnin opiskelu, ohjelmoinnin peruskurssi, keskeyttämissyy, opetus-opiskeluprosessi, opetusorganisaatio

ISBN (painettu) 978-952-248-194-8	ISSN (painettu) 1797-6928
ISBN (pdf) 978-952-248-195-5	ISSN (pdf) 1797-6936
Kieli englanti	Sivumäärä 259 s.

Julkaisija Tietotekniikan laitos

Painetun väitöskirjan jakelu Tietotekniikan laitos

☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2009/isbn9789522481955/

Acknowledgements

This thesis is an end product of an interesting and educative journey. Little did I know what was in front of me when I first started doing research in multi-disciplinary area. When I started working at Helsinki University of Technology, computer science education was a new research field for me. Gradually I became familiar with the field, the way people with a technical background think about teaching and learning, and other researchers to collaborate with. These were my first steps in becoming a member of the community of researchers in this field.

This journey would not have been possible without the guidance from my supervisors and the support from my friends and family. First I want to thank Professor Lauri Malmi who has been my supervisor. His genuine motivation to develop teaching has made it a joy to work with him. He has provided insightful comments, helpful advice and encouragement at all phases of this thesis project. Most importantly, he has introduced me to many members of the community of computer science education researchers. Conference trips and working group meetings broadened my perspective and helped me to build a valuable network. These experiences have been essential in my professional development.

I also thank Professor Emeritus Veijo Meisalo for supervising my thesis project. He provided the much-needed educational viewpoint to the discussions we had during the meetings. Getting feedback and guidance from an experienced educationalist was especially important for me since I worked and collaborated mainly with computer scientists and engineers. He reminded me of the importance of looking at things in a broader context and gently kept on pushing me to develop my thoughts and ideas further. I also thank the thesis' pre-examiners, Professor Mike Joy and Docent Kalle Juuti, for useful comments that helped to improve this work.

I also want to thank fellow researchers and colleges for supporting my research. Anders Berglund and Arnold Pears from Uppsala University were the first ones to welcome me into the international community of researchers. I thank them for stimulating discussions. Judy Sheard from Monash University also gave me valuable feedback during the writing process. The SÄMA study group provided an important forum where I was able to meet other researchers in Helsinki University of Technology who did multi-disciplinary research. I would like to thank all the members of that group: Professor Ari Sihvola, Kirsti Keltikangas, Anna-Kaarina Kairamo, Johanna Leppävirta, Aura Paloheimo, Eija Zitting, and Pirjo Pietikäinen. I also thank the COMPSER research group members Jan Lönnberg and Juha Sorva for interesting discussions and helping me with the English proof reading of this thesis. Anna Shilova also helped me with the proofreading. Special thanks for the professional proofreading of large parts of the final version go to Ronja Addams-Moring.

I thank my colleagues Liisa Hirvisalo, Mervi Ranta, Henrik Asplund and Päivi Haho for interesting and stimulating discussions during the lunch breaks. Our swift discussions brought a nice break to the workday.

My friends Eija Kiuru, Minna Markkanen, Ilona Visapää and Mirva Alakoskela helped me to keep a balance between work and free time. I thank them for listening to me and getting me out of my office every once in a while.

I thank my brother Markku Kinnunen for insightful comments on my text and ongoing encouragement. At the beginning of my journey as a PhD student, he pointed out to me that Helsinki University of Technology was a place to do interesting research. Without this remark my thesis would have taken another direction altogether. Finally, I warmly thank my parents Hilkka and Pentti Kinnunen for their support and encouragement throughout my studies and the writing of the thesis. Their spurring helped me to stay motivated and pursue my goals.

Päivi Kinnunen
In sunny San Diego, November 7, 2009

Contents

List of Figures

List of Tables

1 Introduction

Helsinki University of Technology educates professionals in several engineering areas: such as electrical engineering, machine engineering and civil engineering. As a part of the basic studies curriculum, most of the students are required to take an introductory programming course. The basics of computer programming are considered one of the compulsory skills that, for instance, a Master of Science in Electrical Engineering should possess. However, for several years the introductory programming course for non-CS majors has had a high drop-out rate. This is a relevant problem since for most of the drop-outs, this course is compulsory and therefore drop-outs need to re-enter the course some other year to in order to graduate. In addition, for large-scale courses the high drop-out rate means that there are hundreds rather than dozens of drop-outs. It is a waste of both the students' and the university's resources when these students re-enter the course until they get a passing grade. This concrete problem was the origin of the initial research question *"Why do students drop out of the introductory programming course?"* The process of looking for an answer to this question resulted in both empirical contributions and the development of analytical models.

I started to map the possible reasons for dropping out by collecting empirical data and reading literature. Helsinki University of Technology is not alone with the problem of high drop-out rates in introductory programming courses. Many institutes report drop-out rates of 20-40 percent and multi-institutional studies indicate that there are deficiencies in the learning outcomes of students who have passed one or two computer science courses (Lister et al. 2004; McCracken et al. 2001; Tenenberg et al. 2005). The literature suggests that the factors that contribute to students' success and to their decisions to drop out are manifold. It is suggested that students' attitudes, expectations and experiences of educational methods play a larger role in students' success than their prior academic achievements or skills (Boyle et al. 2002). Studies on reasons to drop out reveal that aspects, such as, professional, academic, financial, personal and satisfaction factors may affect the students' decision (Bennett 2003; Xenos and Pintelas 2002). At the course level, the difficulties the students face may affect the quality of the learning outcomes or contribute to the decision to drop out of the course. The first year university students who take computer science courses, such as an introductory programming course (CS1) in Java, face difficulties that originate from the subject. There are several studies concerning difficult topics in a CS1 course. However, many of the studies seem to focus on topics the teachers find difficult for students to learn in CS1 courses. The students' opinions seem to be studied less often. The Robins, Rountree and Rountree (2003) study serves as a summary of difficulties. The study concluded, based on an extensive literature review, that loops, conditionals, arrays and recursion were those language features that were the most difficult for the students. Likewise, the study also stressed that problem solving, design and expressing a solution/design as a program were issues that students had difficulties with.

As soon as the first empirical and literature review based results started to emerge it became clear that the drop-out phenomenon is rather complex. The combinations of factors that contributed to the students' decisions to drop out seemed to be various and unique to each student. Furthermore, students had several reasons for dropping out: some reasons were related to the course, some to the student, and finally some were related to the university as a larger organisation. This observation made it clear that a broader perspective to the phenomenon was necessary. First, the drop-out phenomenon expanded to difficulties during the instructional process. While some difficulties are severe enough to make the student to drop out, other difficulties just may make the students' life harder. However, which difficulties are critical when it comes to dropping out, varies from student to student. Second, it was no more sufficient to limit the observations to the students' point of view. In order to understand the drop-out phenomenon, I needed to add the teachers' and the university's points of view to the analysis. This three-fold holistic approach to the difficulties and challenges in an introductory programming course became the base to the whole research project. More precisely, the general research goal of this project grew into the following:

Which aspects cause the most difficulties during the instructional process from the students', the teachers' and the university's points of view?

There are some concepts that are used frequently in this study. The concept of the *instructional process* is defined as "*a wide concept consisting of all the important components that take place in classroom instruction as well as the steering factors defined in the curriculum. ... the instructional process is a totality that includes both the teacher's and the pupils' actions as well as learning*" (Kansanen 1999, pp. 86). Even though Kansanen's definition refers originally to the instructional processes that take place in a comprehensive school, in this research this definition is used in a context of higher education. As the definition suggests, instruction is a complex phenomenon that cannot be understood without considering the different actors and the process of instruction in a context. In this thesis, the concept of *university* refers to an institution of higher learning and research. In the empirical part of the thesis, university refers to Helsinki University of Technology (TKK). The concept *teaching organisation* is also used as a synonym for university in places where the emphasis is more on the administrative aspects of the university. One of the chosen point of view concerns students' studying. The concept *studying* is deliberately chosen over *learning* to emphasise the actions the students take to learn. The concept of learning was interpreted to have a connotation that refers more to mental learning. Since the focus of this study is on the actions, studying was regarded as a more appropriate concept.

There were two possible theoretical frameworks that provided the tools to tackle the instructional process as a whole: The General System Theory (GST) (von Bertalanffy 1972) and the Activity theory (Engeström 1987). The activity theory has been used to analyse the learning activities as well as complex working life situations and change processes. The Activity theory focuses on analysing the action itself. The strength of the theory is that it highlights the mediated nature of the actions, the connection between the elements of the activity and the possible conflicts between the elements. Since the theory works on a high abstraction level, it has the potential to tackle many different types of phenomenon. Activity theory has been used, for instance, to receive a holistic picture of the computer science students' experiences (Berglund 2005) and to develop work actions and organisations (Engeström 1990; Engeström 1999). However, in this study I chose to use the General System Theory as a theoretical framework. The decision was based on the GST's ability to tackle the process nature of instruction. Whereas the activity theory's focus is more on the activity itself, the GST focuses more on the process (Haho unpublished manuscript). Using the GST's terms, I am looking at an open system (the instructional process) that receives input (e.g., students) and produces output (e.g., a high drop-out rate) (Figure 1). There are several inputs and outputs that are involved with the instructional process. This study received its motivation from one unwanted outcome - a high drop-out rate. At the beginning, it was not well known why the instructional system produced undesirable outcomes. The instructional system is like a black box that does not allow the observer to see what happens during the process or how the different aspects of the process relate to each other and affect the outcome. The previous instruction related studies have shed light on some aspects of the black box. The present study aims at shedding further light on the instructional process from three different points of view thus providing a holistic approach.

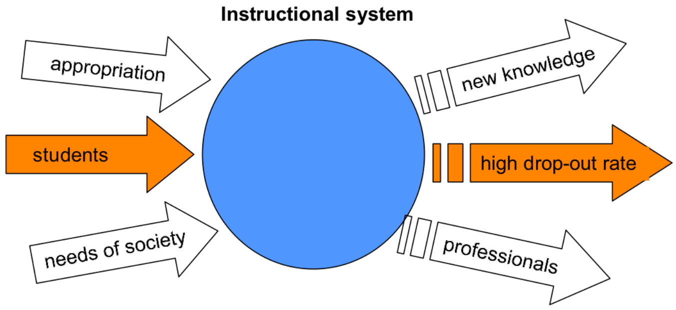


Figure 1 The instructional process as an open system

This study has its roots in two disciplines. On the one hand, its theoretical roots and the developed analysis models are related to Educational science, more precisely to didactics. The basic tenets of general didactics can be clearly seen in the holistic approach that is adopted in this study as well as in the ambition to build models of the instructional process to gain better understanding of the complexity of teaching and studying processes. The other root of this study lies in computing education. The empirical data for this study was collected from the students and the teachers who relate to the computer science course in a university, in many cases to an introductory level programming course. Therefore, this study combines two perspectives. Educational science provides the theoretical framework, and computing education offers the data and, more importantly, a relevant and meaningful research goal.

The general research goal was divided into three main research questions. The first research question focuses on the difficulties and challenges students face when studying computer programming. Moreover, difficult course content, students' strategies to overcome difficulties, and factors contributing to the students' decision to drop out are discussed. The second research question considers computer science teachers' conceptions of the students' studying process and their own teaching process. The third research question tackles the teaching organisation's point of view to the instructional process and provides an insight to the boundaries and possibilities the university, a multi-level organisation, gives to the teachers and students.

I tackled the research questions by developing analysis models as well as collecting and analysing empirical data. The analysis models that were developed for this research are:

- The "dimension doughnut"
- The three-layered didactic triangle
- The feedback loop model

The "dimension doughnut" was created to emphasise the many points of view from which the instructional process can be analysed. Each dimension stands for one possible point of view, for example, the teaching organisation or the type of interaction that takes place in a lecture hall. Additionally, each dimension divides into categories that shed light on the variation within one dimension. The many dimensions that themselves include several categories have the potential to tackle complex phenomena. However, the "dimensional doughnut's" ability to describe the interaction between the dimensions is limited. Therefore, additional analysis models were needed.

The second analysis model was the three-layered didactic triangle[1]. The existing didactic triangle was further developed (see, e.g., Kansanen 2003) into a form that has not been presented before. In the new developed triangle there are three layers: the course level, the teaching organisation level and the society level. The three-layered triangle made it possible to broaden the viewpoint to the organisation and society level processes that the original triangle did not support. In addition, the three-level triangle may be used when the interaction between different levels is analysed. Both the "dimension doughnut" and the three-level didactic triangle are discussed in chapter four.

In this study, the three-layered didactic triangle was also used as a starting point when the new categorisation system for computer science education research was developed. The analysis of the existing literature in the field of computing education research called for an analysis model, which the meta-analysis of the literature could be based on. The previous categorisations, concerning the literature, have been based on what has been studied so far. No analysis has been done based on the didactic aspects of the papers. There are no analyses of which areas are lacking research and which areas would be interesting in the sense of diversifying teachers' and researchers' understanding as well as furthering the field of computing education research. The new categorisation system that was developed in this study is based on the didactic focus of the research. The used categories were derived from the three-layered didactic triangle. The analysis of the research on difficulties students and teachers faced in an introductory programming course highlighted that a much of the literature has concentrated on course level issues, and furthermore the focus is often on aspects that can be expressed quantitatively. The new didactic focus based categorisation system is introduced in chapter four.

The third analysis model that I developed further for this research from the existing instructional process model (Meisalo 1985) was the feedback loop model. The further developed model was based on general system theory. The new model highlights the process nature of instruction and the centrality of feedback in process' self-reproduction. This model enabled analysis of the different phases of the instructional process and the origin and the role of the feedback at each phase. In the empirical part of this research, I also used the feedback loop model to analyse the empirical data. The strengths of the model are that it can be used to analyse the instructional process from the student's, teacher's, and organisation's point of view. In addition, the model also made it possible to highlight the interaction between different points of view, thus revealing some sources of difficulties that students and teachers faced. The model of the instructional process is introduced in chapter five.

Each of the three analysis models that were developed for this study has its strengths and limitations. The "dimension doughnut" highlights the multiple points of view and the richness of the reality. It can be used, for instance, to analyse the variety of student's cognitive processes or the variety of pedagogical actions the teacher can choose from. The limitation of the "dimension doughnut" is that it does not consider the interaction between the dimensions and the process nature of the instruction. The three-layered

[1] The three-layered didactic triangle is not a geometric shape but a mental model

didactic triangle considers the interaction between actors and goals, but fails to emphasise the variation that the dimensions' categories did in the "dimension doughnut". The feedback loop model brings in yet another aspect of the instructional process that neither of the previous tools were able to tackle – the process nature of instruction. This viewpoint sheds light on the mechanisms the instructional system can use to correct its own functions when trying to achieve its goals. The feedback loop model allows the researcher to take a step back and view the instruction as a part of a larger system. The advantage is that in this way the larger mechanisms, that affect the instructional processes on different levels, can be seen. For example, the system's inability to utilise the feedback it receives may explain some of the outcomes of the instructional processes. All three analysis models have their strengths that complete each other by emphasising different aspects of the instructional process.

The empirical data was collected during the years 2005-2008. I used a mixed method approach where quantitative and qualitative phases alternated. This approach was successful in allowing first to map the unknown factors and then to strive to get an insight into interesting phenomena. I used four data sources in this study: students of Helsinki University of Technology (TKK) who were in an introductory programming course, computer science teachers from 13 universities around Finland, representatives of administration and formal documents that concerned the TKK's educational actions.

The empirical contributions of this study concern the engineering students' and the computer science teachers' complex reality. The results reveal the variety of reasons for dropping out of the course, the difficult topics - they also highlight the organisation or course related factors that generate challenges. The results highlight that there are different types of sources for difficulties and challenges varying from the nature of the course topic to the students' studying skills and the restrictions the teacher faces in a large-scale course. The changing combinations of the difficulties make it a challenging task to address them efficiently.

The empirical contributions of this study are:

- Knowledge of why students drop out of an introductory programming course (CS1). The results highlight the manifold and cumulative nature of reasons behind the decision.
- Knowledge of which content related aspects students who pass the introductory programming course and students who drop out of the course, find difficult to learn.
- Knowledge of which strategies these students find helpful when they face difficulties in a CS1 course.
- Knowledge of computer science teachers' conceptions of studying and teaching. The results highlight, e.g., in which aspects teachers think students have difficulties and how teachers think they can affect the studying.
- Knowledge of how some education quality related statements were expressed in different formal documents at TKK and how the teaching organisation gave only general resolutions to guide the Faculties' and Departments' actions.
- Knowledge of the challenges the representatives of administration at TKK see in the instructional process. For instance, the collection and utilisation of feedback was not systematic, and the setting of goals and planning were content oriented.

- Knowledge of how the students', teachers' and organisation's instructional processes interrelate: what kind of information flows or does not flow between the students', teachers' and the teaching organisation's levels.

The empirical results of this study are reported in three chapters. The students' difficulties during the course and their reasons for dropping out are presented in chapter 7, the teachers' conceptions of studying and teaching are reported in chapter 8. The results concerning the organisations actions are presented in chapter 9.

In summary, the thesis is organised as follows (see also Figure 2). The first two chapters provide the background for the research. The second chapter includes the literature review on difficulties and challenges students and teachers confront in an introductory programming course. Chapter 3 introduces the research questions. Chapters four and five describe the analysis models that were developed for this study. These models are a part of the analysis methods that were used in later phases of the study. In chapter six I discuss the procedure and quality of the empirical part of the study. The empirical results of the study are presented in chapters seven, eight, and nine. Chapters ten and eleven conclude the thesis.

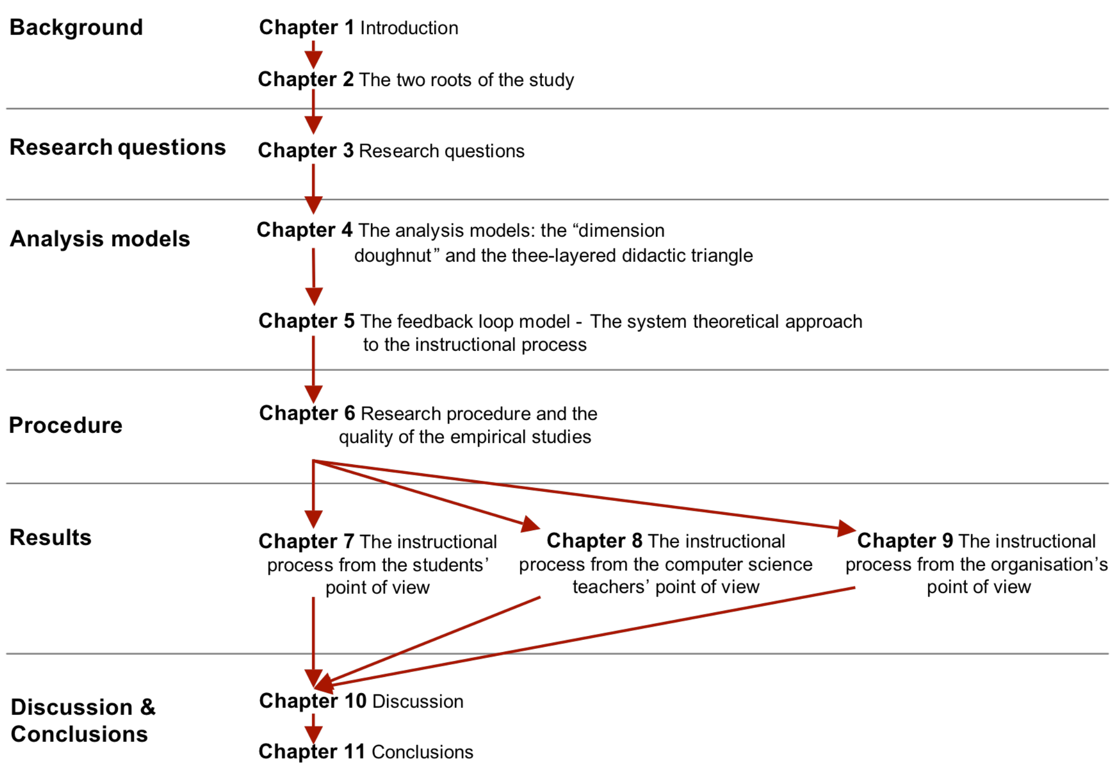


Figure 2 The structure of the thesis

The initial research question, "Why do students drop out of the introductory programming course?" originated from one undesirable outcome, the high drop-out rate in an introductory programming course. However, when this outcome was placed in a larger context the multi-faceted nature of the instructional process and the reasons behind drop-out rates unfolded. The "bird's eye view" on the initial research question resulted in a holistic understanding of the instructional process, which helps not only to tackle the drop-out rates but also to enhance the instructional process on several level.

2 The two roots of this study

This study places itself on the intersection of two disciplines. On the one hand, it has its roots in Education and, on the other hand, in Computing Education (CE); particular emphasis is given to Computer Science Education (CSE). Education provides the base for the theoretical part of the research whereas the concrete research questions are derived from computing education. This chapter discusses both of these essential foundations. The first part of the chapter discusses how this study is placed in an educational field as a part of subject-matter didactics. The second part of the chapter focuses on the computer science education. The third part of the chapter introduces the related literature.

2.1 One foot in Education

General education is a wide-ranging discipline that has diverged into several sub-disciplines over the years. Traditional classification, for example, Heinonen (1989) has categorised Finnish Education into eight sub-disciplines: philosophy of education, anthropology of education, didactics, psychology of education, sociology of education, comparative education, history of education, and special needs education. Furthermore, for example, didactics can be classified into sub-categories, such as, general and subject matter didactics, or theoretical and normative didactics. Another possible approach to the classification of Education is to take population (e.g., adult education), subject matter (e.g., mathematics), context (e.g., school education) or neighbouring disciplines (e.g., sociology) as a basis for the categorisation (Uljens 2001).

The survey over the last six years' paper sessions during the annual Finnish conference in educational research arranged by FERA (Finnish Educational Research Association) revealed that Heinonen's classification is still valid and research has been done in all eight sub-fields. Furthermore, there has been a wide range of research focusing on different school levels from early childhood education to higher education. (FERA 2003; FERA 2004; FERA 2005; FERA 2006; FERA 2007; FERA 2008) Similar classification of research areas was found by surveying strands at the European Conference on Educational Research that is arranged by the European Educational Research Association (EERA). (EERA 2006; EERA 2007)

In comparison, the American Educational Research Association (AERA) has classified the field of educational research into 12 divisions that represent broad substantives or professional interests. The earlier mentioned sub-fields of Education (Heinonen, 1989, EERA 2006; 2007) were partly found in AERA's classifications, too. However, some fields were not given such an independent status. For example, the philosophy of education did not exist as its own field of a classification title level. In addition, continental researchers have understood didactics slightly differently than Anglo-American researchers. In Anglo-American countries didactics was found in under several division titles such as Curriculum Studies, Learning and Instruction, and School Evaluation & Program Development (AERA 2007). The difference in classification and emphasis of subfields in the Continent and the America is a matter of different cultures that have their own philosophical, and political roots. Uljens (2001, pp. 295), for example, stated that "*From an American perspective it may seem odd to have several sub-disciplines in education. From a Nordic perspective again it is odd that education* *is not an autonomous discipline at every American university, but is instead conceived of as a "a field of research".*"

This thesis belongs to the field of didactics, subject-matter didactics, and more specifically to the computer science didactics. The aim of the thesis is to discuss the instructional process that includes teaching, studying and learning in a context of computer science education (CSE) at higher education.

Didactics can be defined as a sub-area of Education that discusses all instructional process related questions. Especially, it discusses the whole process of instruction including branches, such as, educational policy and curriculum theory. Therefore, didactics has a common interest with educational psychology, which is defined as an intersection of education and psychology. Since the number of publications in the area of computing education research (CER) and computer science education research (CSER) are influenced by educational psychology, it is advisable to clarify the differences between these disciplines.

According to Kansanen (2002), in Germany and Nordic countries educational psychology and didactics exist as separate sub-disciplines of education. However, didactics is virtually non-existent in Anglo-American countries, and instead, there is a stronger emphasis on educational psychology. Kansanen stated that there are several differences between didactics and educational psychology. For example, the goal of didactics is to build models or systems for the instructional process in society. Educational psychology's mission, in contrast, aims at developing theory in the area of educational psychology. Didactics concerns the achievement of the aims and goals of some curriculum. That is, didactics discusses the context-dependent issues that take into account the values and goals of curriculum. On the other hand, educational psychology concerns student characteristics, motivation and learning in more context-free situations, taking the general value basis in the society as a background. Furthermore, the object of attention in didactics is the instructional process as a whole whereas in educational psychology it is some special topics of the instructional process. For example, educational psychology is interested in questions, such as, what kind of attitudes students have towards mathematics. Didactics is interested in the content, goals, and processes of the instruction, as well as their interrelationships.

The study by Seel (1999) outlined three theoretical branches for didactics: theory of Bildung[2], theory of teaching, and theory of school. This categorisation further highlighted the didactics' holistic framework in contrast to educational psychology. However, no clear line can be drawn between these two disciplines. The difference is more based on the degree, for instance, of context-dependency than clear-cut definitions. The essential aspect from this thesis' point of view is to be aware of the two different sub-disciplines of education that will help to define the context of this study.

Subject-matter didactics can be understood as a more focused instance of didactics. It shares the same framework with didactics but focuses on teaching and learning of a specific subject, such as, computer science (Seel 1999). Even though didactics has far reaching roots that date back to the mid 17th century, the subject-matter didactics has a much shorter history. For example, the study by Bertrand and Houssaye (1999) stated that in the Francophone world the scientific interest in subject-matter didactics dated back to the seventies. Around the same time in Finland, education of teachers was

[2]Bildung does not directly translate to English. Here it can be understood as general education or all-round education.

transferred to the universities, resulting in rising scientific interest towards subject-matter didactics. Computer science is a fairly new discipline and thus the history of computer science education research is short.

The interest in technology education is not a new phenomenon. For example, the American Society for Engineering Education (ASEE) was founded already in 1893. On the other hand, the European Society for Engineering Education (SEFI) was founded in 1973. The conferences that are arranged by these societies consider a vast variety of engineering education related topics, such as issues arising from computing education.

As the computing and technology education fields expanded, specific organisations and conferences were arranged. The International Federation for Information Processing (IFIP) was established already in 1960. The First ACM Technical Symposium on Computer Science Education (SIGCSE) was held in 1970 and The First Annual Frontiers in Education (FIE) Conference in 1971. Sixteen years later, the Psychology of Programming Interest Group (PPIG) was established. The First Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE) was held in 1996 as well as the First Australasian Computing Education Conference (ACE). The First Baltic Sea Conference on Computing Education Research (Koli Calling) was held in 2001 and The First International Computing Education Research Workshop (ICER) was held in 2005. The growing number of new international educational conferences is indicative of enlarging the community of researchers and teachers who specially are interested in computing education.

The basic questions that subject-matter education research endeavours to answer or the complex phenomenon that it seeks to understand better are fundamentally the same regardless of the subject matter. For example, the Annual ASEE Global Colloquium on Engineering Education in 2007 has stated common research questions, such as, what is it that enables learning to take place, how do we facilitate it, how do we determine what students have actually learned (ASEE 2007). These questions interest researchers across conferences and subject matters.

As stated earlier, this thesis has its roots in both education and computing education. From education point of view, the context of this study can be visualised as inclusive squares (Figure 3). The context of this study is based on computer science education as a part of computing education, which is an instance of subject-matter didactics. Further, subject-matter didactics is a part of didactics, which is a sub-discipline of education. I have made a distinction between computing education and computer science education knowing that the concepts are sometimes used as synonyms among the researchers in these fields. It seems that recently the term computing education has become more popular in the community of researchers. For instance, computing education is included in the name of the youngest conferences in a field (the Australasian Computing Education Conference, the Baltic Sea Conference on Computing Education Research, the International Computing Education Research Workshop) in contrary to older conferences, such as, the ACM Technical Symposium on Computer Science Education.

I define computing education as an extensive concept that includes a variety of research topics ranging, for example, from the teachers' perceptions of learning technologies (Cope and Ward 2002) to the importance of conceptual models when learning the software artefacts (Ben-Ari and Yeshno 2006). I define computer science education (CSE) as a special sub-division of computing education (CE) that focuses on discussing issues related to learning and teaching computer science as a discipline. The following examples illustrate the difference between the two research fields. Computing education research includes topics from how to learn to use software to how learn to make software; computer science education research focuses more on the latter. Further, studies that could be categorised as computer science education would discuss, for instance, factors that predict students' performance at a programming course (Bergin and Reilly 2006) or students' perceptions of some programming language specific concepts (Eckerdal 2006). The study by Namazi and McClintic (2003) on how elderly persons in a long-term care setting learn to use computers for typing, playing games, sending e-mails, and surfing the web is an example of the study that would be categorised as computing education.

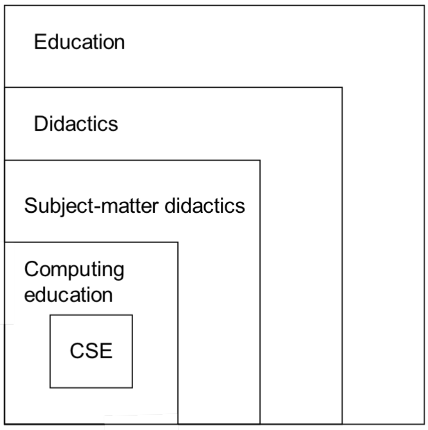


Figure 3 This study in the context of Education

In summary, in this thesis, I define computing education as a larger study field that contains computer science education. The difference between the two is that computer science education limits itself to the aspects that relate to computer science as an academic field as the computing education may discuss some other computing related aspects in addition to computer science[3].

This study discusses computer science education (CSE) also as a part of computing education, which can be understood as a sub-field of computing. See the lower squares in Figure 4. Computer science education research (CSER) serves the context to this thesis in a sense that empirical data is collected from the introductory programming course at a technical university. The aim of this study is to enhance the understanding of the instructional process in this context taking the characteristics of computer science into account. Figure 4 describes how the context of this study can be found in the area where two disciplines overlap.

[3] CSE and CE researchers do not usually make a clear distinction between computing education and computer science education. These labels are often used as synonyms.

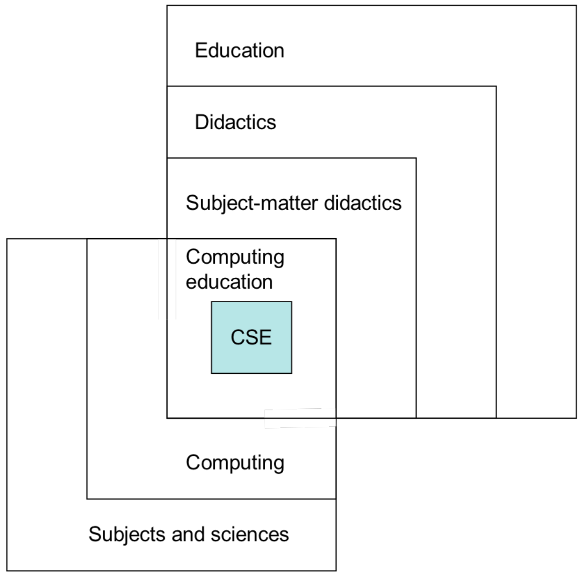


Figure 4 The two contexts of this study

2.2 Other foot in computer science education

Computing as a discipline includes several sub-fields. The ACM Curricula (2005) lists five sub-fields: Computer Science, Information Systems, Software Engineering, Computer Engineering, and Information Technology. Therefore, the computing education research (CER) also may cover a wide range of research topics. It is a cross-disciplinary research area since it comprises of several disciplines including computer science or some other subfields of computing and some other fields, such as, education, psychology of education, and sociology of education. CER discusses a wide range of issues that are related to: students' understanding, teaching practices, curriculum development, gender, and retention. The research in this field is often pragmatic oriented, with research questions emerging from computer science teachers' own experiences. The common orientation is to investigate the feasibility of a teaching method or a tool in order to enhance the students' learning. The computer science education research (CSER), on the other hand, can be understood as a sub-field of computing education research. The objectives of CSER are more focused on computer science. Therefore, since the data of this thesis is collected from the introductory programming course, CS1, this study belongs to the field of CSER.

The usage of concepts computer science education research (CSER) and computing education research (CER) is not yet well established among the researchers and the two concepts are frequently used as synonyms. Therefore, in a following discussion CER and CSER could be interchanged. The choice of the concept in each case depends on which concept the authors have used.

The field of computing education research (CER) has reached a point where there already exist a vast number of publications. However, the field does not yet have an established structure or a widely recognised position as an equally weighted research area amongst CS related research areas. There are some actions taken to define the content and state of the CER. Sally Fincher in Stasko et al. (2001) presented one of the first categorisations for the research done in CSER. Four categories included: small-scale investigations and the scope of study, tool motivated investigations, mental and conceptual related investigations, and finally the research paradigm. This categorisation seems to have been influential when subsequent categories have been created.

Fincher and Petre (2004) categorised ten research areas in computer science education research. The main categorisation base was the topic of the paper even though the nature and the scope of the paper were also discussed. In the study by Pears, Seideman, Eney, Kinnunen, and Malmi (2005) authors used categories that were introduced in the paper of Fincher and Petre as a starting point for their analysis. Furthermore, the authors added another dimension that highlighted the significance of the papers to the field. Valentine (2004) introduced the viewpoint that takes the character and the content of the paper as a basis for categorisation. The studies by Simon (2007) and Simon, Carbone et al. (2008) and Sheard et al. (2009), on the other hand, applied a classifying system that had four dimensions: nature of the paper, topic, context, and the breath of paper's context. The study by Glass et al. (2004) based the analysis of publications in the computing disciplines on the topic, the research approach, the research method, the reference discipline, and the level of analysis. The meta study by Randolf et al. (2008) of methodological properties of publications in the filed of CSE draw the attention, for instance, to research methodology, research design, and variable examined. The thesis of Berglund (2005) adopted another approach entirely, and made the distinction between studies in CER based on the educational tradition to which publications relied on. The variety of categorisation system illustrates the researchers' interest in attempting to define the content, context, and standards for quality of the new research area. The desire to define the content and the state of CER is something that reflects its similarity with other fields (see, for example, papers concerning mathematics education (Niss 1999) and the aim of educational research (Hostetler 2005), and higher education research as tribe (Tight 2008)). The following list presents the ten research areas identified by Fincher and Petre (2004). The division of the four main areas (A, B, C, and D) is based on the work of Pears et al. (2005).

A) Studies in Teaching, Learning, and Assessment

This heading covers studies that discuss students' characteristics, actions and understandings, along with teachers' actions in different pedagogical settings. Many papers in this category seek to find a better or proper way to enhance students learning in various pedagogical settings. The focus of the papers is often on some special topic or conception. Three of Fincher's and Petre's (2004) categories belong to this area:

- Studies of *student understanding* investigate the students' mental and conceptual models along with their perceptions and misconceptions.
- *The teaching methods* subfield includes, on the one hand, studies that investigate how teachers can help students learning and, on the other hand, how teachers can affect and control the teaching interaction in order to make interaction more effective.
- Research papers that fall in to the field of *assessment* discuss the different types of assessment, the validity of assessment, and the automating grading.

B) Institutions and Educational Settings

This area contains papers that discuss the social and institutional context in which education is provided. There is a wide range of topics that are included in this area that range from minority issues to curriculum studies. The following four categories identified by Fincher and Petre (2004) fall into this area.

- Studies on *transferring professional practice into the classroom* seek to help introduce professional practices into the classroom in order to enhance students' understanding and to help students prepare themselves for the transition into the professional workforce.
- Studies in *transferring from campus-based teaching to distance teaching* contemplate on issues that are faced when classroom teaching is transitioned to e-learning/remote learning.
- *Recruitment and retention* subfield includes studies that discuss the gender and diversity issues as well as issues that contribute to students' performance in computer science.
- Studies in the *construction of the discipline* subfield discuss issues, such as the nature and the content of the CER discipline, along with curriculum issues.

C) Problems and solutions

This heading covers studies that concentrate mainly on the tools and technology that have the potential to help with some aspects of learning computing. Studies that discuss usage or devising new tools relate to the larger research area that takes into account modern learning environments. Three of categories identified by Fincher and Petre (2004) belong to this area.

- Studies concerning *animation/visualisation/simulation systems* are interested in changes in teaching and learning when different systems are used.
- *Educational technology* includes studies on developing and applying new technologies or environments to enhance learning.
- Studies in the area of *incorporation of new development and new technologies into classroom* are closely related to transferring professional practices into the classroom.

D) CER as a discipline

The area D is unique for the study by Pears at al. (2005) in a sense that it does not contain any categories identified by Fincher and Petre (2004). The paper of Pears et al. stated that topics of the papers in this category included those that contribute to the development of research methodologies for CER. Therefore, papers that introduce a theory or a research approach to the community of CER researchers belong to this area. Examples of papers that could fall into this category are papers on constructivism (Ben-Ari 2001), social construction (Machanick 2007), situated learning (Ben-Ari 2004; Ben-Ari 2005) and phenomenography (Berglund 2006).

Summary

Using the classification of Fincher and Petre (2004), this study could be classified partially into teaching methods, student understanding, assessment and construction of the discipline. However, this study's approach differs notably from the one adopted by majority of other studies with similar classification. Even though the study discusses students' and teachers' actions in pedagogical settings, this study aims at looking at them from the instructional process's point of view. The utilisation of feedback is a central aspect of goal-oriented processes and therefore, an assessment of this sort of feedback is relevant to this study, too. This study also discusses the instructional process from the organisation's point of view, touching upon the feedback mechanism that affect the content of the curriculum.

2.3 Difficulties during the instructional process - literature review

The general research goal of this project is to highlight which aspects cause difficulties during the instructional process. The origin of this goal was to understand the reasons for the high drop-out rate in CS1 course. Moreover, the viewpoint has widened to cover the challenges that do not necessarily cause students to drop out of the course, but still, nevertheless, may affect learning outcomes. The adopted holistic viewpoint is further enriched by adding teachers' and teaching organisation's points of view to the difficulties encountered in studying and teaching computer science.

The challenges and difficulties during the instructional process are a vast. The following literature review emphasises the variety difficulties that are reported in the field of computer science education. The main focus is further limited on the difficulties that are detected in introductory programming courses at the university level. However, the papers in the field of computing education that have a larger focus are also included if the research highlights an aspect that is studied less in the context of introductory programming.

One essential aspect in defining the relevant literature is to specify the indicator of difficulties in the instructional process, and defining difficulty in the first place. The consequence of the difficulty is then taken as a criterion. The factors are regarded as difficulties if they negatively affect the instructional process. Naturally, the difficulties vary by their type, severity, and the way they emerge. For example, dropping out of the course or a degree program clearly indicates that there are some severe problems and thus the reasons for dropping out tell something about the difficulties. On the other hand, students' misconceptions on some critical concepts may not be the reason for dropping out, yet they certainly affect the quality of the learning outcomes. In this literature review, the focus is limited on the studies that:

- Discuss the difficulties during the instructional process in an introductory level computer programming course at a university.
- Either analyse some particular difficulty or discuss how some earlier defined difficulty has been addressed (thus revealing that the difficulty is perceived severe enough that it must be addressed).
- Analyse the drop-out phenomenon, high attrition rates and the retention in relation to a single course or a whole degree program. The papers that discuss the whole degree program may shed light on some of the challenges and difficulties that occur during single courses, and thus these papers are included in this discussion.
- Sheds light on the variation of the types of challenges and difficulties faced during the instructional process.

The main criteria for including the difficulties and success factors in this analysis were that they had been verified by an empirical study. However, few papers raised factors that were anecdotal by nature. In one or two cases, these factors were included since they highlighted a relevant aspect from this study's viewpoint and thus contributed to the analysis by broadening the pool of difficulties and success factors. The variety of the type of difficulty is the prime focus of this review and the degree of difficulty is a secondary aspect. For instance, several studies demonstrated which course content were difficult to learn. In many reports the course topics were listed in order of difficulty. This review is not interested in what was the most difficult concept to learn. Instead, it concentrates on the more general aspect of the difficulty.

This literature review includes both conference publications and journal articles. The most of the conference papers that are included into this literature review were published in International Computing Education Research Workshop (ICER), Baltic Sea Conference on Computing Education Research (Koli Calling), and Australasian Computing Education Conference (ACE). The majority of the journal papers were published in SIGCSE journal and Computer Science Education. Overall, conference papers were in slight majority in this literature review.

The following paragraphs are organised so that first the course content related difficulties are discussed from students' (Table 1) and teachers' viewpoints (Table 2). Then, other related difficulties are discussed, respectively from the two different viewpoints (Table 3 and Table 4). The categorisation of the difficulties in Table 3 and Table 4 results from analysis of the research. Finally, a summary of the difficulties revealed by the literature review is presented.

2.3.1 Course content related difficulties students confront

It is long recognised that learning programming is a challenging task. The students in an introductory programming course seem to face several challenges during the short period of time. Already du Boulay (1986) highlighted five partly overlapping areas of difficulty that a novice programmer faces: First there was *a problem of orientation*. Du Boulay listed aspects, such as, finding out the advantages of learning programming and realising what kind of problems can be tackled with programming to this category. Second, there were difficulties related with *the notional machine*. These problems were related to the understanding the general properties of the computer and the relation between the physical machine and the notional machine. Third, there were difficulties that relate to *the notation of the formal languages* including aspects, such as, mastering the syntax and the underlying semantics. Fourth, du Boulay discussed the difficulties of *acquiring structures*, such as, loops. The fifth difficulty concerned *the pragmatics of programming*, that is, the skill of specifying, developing, testing and debugging a program. In the following paragraphs, the five areas of difficulty that du Boulay listed are used to classify the content related difficulties found in the literature. The study by Schulte and Bennedsen (2006), which based its categorisation on course topics on du Boulay's areas, has been used as a guideline in the categorisation of the topics.

1. Problem of orientation

Du Boulay's (1986) definition of the problem of orientation included aspects such as understanding why it is worthwhile to learn programming, along with kind of tasks that can be tackled using programming. The essence is thus to understand the general idea of programs and what they are good for. The study by Schulte and Bennedsen (2006) included topics, such as, ethics and algorithm efficiency in this category. In the study

Schulte and Bennedsen asked the teachers to rate how difficult it was for students to learn various topics. The results suggested that algorithm efficiency is one of the most difficult topics for students to learn.

2. Notional machine

The paper of du Boulay (1986) defined notional machine related problems as topics that concerned an understanding of the general properties of the computer and the relation between the physical and the notional machine. The difficulties with understanding such properties along with the role of the notional machine may lead to different kinds of incorrect understandings, thus make learning more challenging. For example, studies by Sorva (2007; 2008) analysed students' understandings of program execution and the notional machine. More specifically, studies have shed light on the ways students understand the concept of variable and object storage in memory. Sorva's studies discussed both students' correct and incorrect understandings providing knowledge of possible sources of difficulties students may face in an introductory programming course.

The study by Robins, Rountree and Rountree (2003) summarised, based on an extensive literature review, many various kinds of difficult topics, ranging from language features to novice programmers' characteristic behaviours and attitudes. One of the sources of difficulties that were discussed in the paper was the novice programmer's tendency to approach programming "line by line".

3. Notation of formal languages

The notation of the formal programming language includes aspects such as mastering the syntax and the underlying semantics (du Boulay 1986). The problems concerning the semantics are long recognised as problematic by several studies. The study by Spohrer and Soloway (1986) analysed the origin of novice programmers' mistakes and provided a taxonomy of novices' problems. One of the levels of the taxonomy included problems related to learning the correct semantics of language constructs. The study by Garner, Haden, and Robins (2005) verified that the syntactic details of the language cause difficulties for the students. The authors analysed the problems students sought help for during an introductory programming course's laboratory sessions, and thus provided the students' point of view to the difficulties. The results disclosed that the basic syntactic details were the most common reason for students to ask for help. The student also asked help concerning data flow. The study by Robins, Haden and Garner (2006) seconds the difficulty of data flow.

4. Acquiring structures

Structures refer to abstract solutions to some standard problems. Several studies have highlighted many structure related aspects that are difficult for the student to learn. Some of the most often mentioned difficult structures were:

- Recursion (Goldman et al. 2008; Lahtinen et al. 2005; Robins et al. 2003; Schulte and Bennedsen 2006)
- Arrays (Garner et al. 2005; Meisalo, Suhonen et al. 2002; Robins et al. 2006; Robins et al. 2003)

- Loops (Garner et al. 2005; Robins et al. 2006; Robins et al. 2003)
- Constructors (Garner et al. 2005; Milne and Rowe 2002; Robins et al. 2003).
- Inheritance (Goldman et al. 2008; Schulte and Bennedsen 2006)
- Polymorphism (Milne and Rowe 2002; Schulte and Bennedsen 2006)
- Advanced data structures (Lahtinen et al. 2005; Schulte and Bennedsen 2006)
- Pointers (Lahtinen et al. 2005; Milne and Rowe 2002)
- Algorithms (Seppälä et al. 2005; Xinogalos et al. 2006)
- References (Lahtinen et al. 2005)
- Using the language libraries (Lahtinen et al. 2005)
- Methods (Meisalo, Suhonen et al. 2002)
- Operator overloading (Milne and Rowe 2002)
- Dynamic allocation of memory (Milne and Rowe 2002)

Many of the studies in the list had asked the teacher the rate the difficulty of the course topic. The studies by Lahtinen, Ala-Mutka, and Järvinen (2005) and Milne and Rowe (2002) took slightly different approach; they asked both the students and the teachers to rate the difficulty of various concepts and topics of object-oriented programming.

5. Pragmatics of programming

According to du Boulay (1986), the pragmatics of programming includes aspects, such as, specifying, developing, testing, and debugging a program. The literature review revealed several studies that shed light on the difficulty of these aspects. For example, the aspects of program design were found difficult by many studies. The results of the study by Garner et al. (2005) disclosed that students faced difficulties relating to understanding the task (what the program is supposed to be doing) and program design (I understand the task but I cannot turn that understanding into an algorithm/a program). The studies by Lahtinen et al. (2005), Spohrer and Soloway (1986), Robins et al. (2003), Garner et al. (2005), and Goldman, Gross et al. (2008) seconded that students found designing a program challenging. The study by Lahtinen et al. (2005) also revealed that dividing functionality into procedures was difficult for the students to understand.

The study by Robins et al. (2003) highlighted that general problem solving skill and expression of the solution or design as a program were difficult for students. Finally, the study by Spohrer and Soloway (1986) highlighted the plan composition problems, which make it difficult to put plans together correctly. This problem was further divided into several sub-categories that included possible problematic aspects such as mapping from natural language to a programming language, and using the previous experiences to develop plans.

Pragmatics that relate closely to the writing and running of the program were also found problematic. For example debugging (Goldman et al. 2008; Lahtinen et al. 2005), exception handling (Goldman et al. 2008), error handling (Lahtinen et al. 2005), and tracing (Goldman et al. 2008) cause difficulties for the students.

The large number of papers, that discussed CS1 course content related difficulties, has affected also the content of the literature that analyses the pedagogical means that are used to enhance students' learning. For example, the study by de Raadt, Toleman and

Watson (2007) considered students' difficulties concerning the programming strategies and reported an experiment that was executed to see if the programming strategies can be taught explicitly. Finally, Table 1 summarises the course content related difficulties and problematic aspects, which were obtained from the literature.

Table 1 Course content related difficulties that students confront

1. Orientation (du Boulay 1986)
   a. Algorithm efficiency (Schulte and Bennedsen 2006)
2. Notional machine: understanding the general properties of the machine (du Boulay 1986)
   a. Students' partially incorrect understandings of the idea of an object stored in memory and the concept of variable (Sorva 2007; Sorva 2008)
   b. Approaching programming "line by line" (Robins et al. 2003)
3. Notation of the formal languages (du Boulay 1986)
   a. Construct-based problems (correct semantics of language constructs) (Spohrer and Soloway 1986)
   b. Mechanical details (e.g., braces, brackets, semi-colons) (Garner et al. 2005)
   c. Data flow (Garner et al. 2005; Robins et al. 2006)
4. Acquiring structures (du Boulay 1986)
   a. Algorithms (Seppälä et al. 2005; Xinogalos et al. 2006)
   b. Arrays (Garner et al. 2005; Meisalo, Suhonen et al. 2002; Robins et al. 2006; Robins et al. 2003)
   c. Loops (Garner et al. 2005; Robins et al. 2006; Robins et al. 2003)
   d. Constructors (Garner et al. 2005; Milne and Rowe 2002; Robins et al. 2003)
   e. Methods (Meisalo, Suhonen et al. 2002)
   f. Recursion (Goldman et al. 2008; Lahtinen et al. 2005; Robins et al. 2003; Schulte and Bennedsen 2006)
   g. Inheritance (Goldman et al. 2008; Schulte and Bennedsen 2006)
   h. Polymorphism (Milne and Rowe 2002; Schulte and Bennedsen 2006)
   i. Advanced data structures (Lahtinen et al. 2005; Schulte and Bennedsen 2006)
   j. Operator overloading (Milne and Rowe 2002)
   k. Pointers (Lahtinen et al. 2005; Milne and Rowe 2002)
   l. References (Lahtinen et al. 2005)
   m. Dynamic allocation of memory (Milne and Rowe 2002)
   n. Libraries (Lahtinen et al. 2005)
5. Pragmatics of programming: Specifying, developing, testing and debugging a program (du Boulay 1986)
   a. Understanding the task (Garner et al. 2005)
   b. Problem solving (Robins et al. 2003)
   c. Design (Garner et al. 2005; Goldman et al. 2008; Lahtinen et al. 2005; Robins et al. 2003; Spohrer and Soloway 1986)
   d. Expressing a solution/design as a program (Robins et al. 2003)
   e. Dividing functionality into procedures (Lahtinen et al. 2005)
   f. Putting plans together correctly (Spohrer and Soloway 1986)
      i. Mapping from natural language to a programming language (Spohrer and Soloway 1986)
      ii. Using the previous experiences to develop plans (Spohrer and Soloway 1986)
   g. Debugging (Goldman et al. 2008; Lahtinen et al. 2005)
   h. Exception handling (Goldman et al. 2008)
   i. Error handling (Lahtinen et al. 2005)
   j. Tracing (Goldman et al. 2008)
   k. Programming strategies (de Raadt et al. 2007)
   l. Applets and graphics (Meisalo, Suhonen et al. 2002)

2.3.2 Course content that teachers find difficult to teach

All the research papers that were discussed in the previous section demonstrate the content related difficulties that students may face during the introductory programming course. However, teachers' experiences of which course content are difficult to teach are a less studied area. The study by Dale (2006) asked the teachers to list topics, which they found difficult to teach. The paper presented four large categories: problem solving and design, general programming topics, object oriented constructs, and student maturity. The problem solving and design category included three subcategories, concerning with general problem solving, algorithm design/development, and object-oriented problem solving and design. The general programming topics category included topics such as parameters, arrays, recursion, pointers, loops, files and conditionals. In addition, testing and debugging were considered difficult to teach. Object oriented constructs category included topics such as polymorphism, inheritance, instance methods, instance variables, and static variables. The fourth category, student maturity, will be discussed later since its content exceeds that of the course content.

The study by Carbone, Mannila, and Fitzgerald (2007) highlighted the aspect of *domain complexity* as a source for unsuccessful teaching. The respondents of the study mentioned the abstract nature of the discipline as one reason for unsuccessful teaching situations. Table 2 summarises the content related aspects that the teachers found difficult to teach according to the literature review. The literature review did not reveal difficulties related to the orientation (the first of du Boulay's categories).

Table 2 Course content related aspects that teachers found difficult to teach

2. Notional machine: understanding the general properties of the machine (du Boulay 1986)
    a. Instance and static variables (Dale 2006)
3. Notation of the formal languages (du Boulay 1986)
    b. Parameters (Dale 2006)
4. Acquiring structures (du Boulay 1986)
    c. Arrays, loops, pointers, (Dale 2006)
    d. Recursion (Dale 2006)
    e. Polymorphism, inheritance (Dale 2006)
    f. Files and conditionals (Dale 2006)
5. Pragmatics of programming: Specifying, developing, testing and debugging a program (du Boulay 1986)
    g. Problem solving (Dale 2006)
    h. Algorithm design/development (Dale 2006)
    i. Object-oriented design (Dale 2006)
    j. Testing (Dale 2006)
    k. Debugging (Dale 2006)

- Domain complexity (Carbone et al. 2007)

2.3.3 Non-course content related difficulties students confront

The following paragraphs concentrate on factors other than course content/topic related difficulties students may face during the introductory programming course. The following paragraphs are structured so that first the factors that may contribute to the attrition and retention are discussed. Then the studies that analyse success factors are discussed. Finally, the difficulties are gathered into Table 3.

Reasons for dropping out of an introductory programming course

The study of informatics course in an open university conducted by Xenos, Pierrakeas, and Pintelas (2002) concluded that there were five main reasons for dropping out of the course: professional, academic, family, heath-related, and personal reasons. The most often mentioned reason for dropping out was the students' inability to estimate the time required for their profession which resulted too little time for studies. The academic reasons included aspects such as the lack of confidence to pursue university-level studies and the lack of assistance from the tutor. The study by Meisalo, Suhonen et al. (2002) seconded the importance of the role of the time managing in the drop-out phenomenon. The study analysed the reasons for dropping out a web-based introductory programming course (CS1). According to the results nearly 45% of the students dropped out of the course due to the lack of time. A fifth of the students dropped out because they found the exercises too difficult. The study by Meisalo, Sutinen, and Torvinen (2002) continued in framework of the previous study and proposed some new factors that may have influenced the drop-out rates. The authors highlighted that students' preconceptions and knowledge about how demanding the course is along with the tutor-teachers' ability to inform students about arrangements and the course may have played a role in the drop-out phenomenon. In addition, tutor-teachers' negative attitudes towards, for example, giving advice, correlated to the percentage of drop-outs.

The study by Beaubouef and Mason (2005) discussed some of the possible reasons for high attrition rate among CS majors. The paper highlighted student related factors such as poor math skills and problem solving abilities, and poor project management skills. In addition, the paper drew attention on teaching institution related factors. For instance, poor advising before and during studies in college, poorly designed CS1 lab courses, lack of practice/feedback during the CS1 course, graduate student teachers (e.g., no training for teaching), and the choice of language and object early vs. object late approach.

Factors predicting success in an introductory programming course

There is a large pool of literature that has analysed the factors that possibly contribute to the students' performance in introductory-level programming courses. However, how the authors' approach success vary. For example, some have looked at the success across the whole curriculum while many others have concentrated on analysing success in a single course. Studies on students' success on an introductory programming course have focused on mainly three factors: students' previous academic experiences (especially programming and mathematics experiences), cognitive and psychological factors, and learning skills (Bergin and Reilly 2006). At the same time as these studies highlighted the possible success factors they also shed light on the possible sources of the difficulties students may face during the introductory-level programming courses. Therefore, these studies are included in the analysis.

The relation between students' prior academic experience and success in an introductory programming course or degree level studies has interested many researchers. Several studies agree that the prior achieved mathematical skills contribute to the success in CS1 courses (Bennedsen and Caspersen 2005; Bergin and Reilly 2006; Byrne and Lyons 2001; Cantwell Wilson 2002; Pioro 2006). However, opposite views have stressed the unimportance of mathematics background in relation to success in CS1 (Boyle et al. 2002; Ventura 2005). Contradictory results concerning the role of the prior achieved mathematics skills may be a result of several variables, such as the type of prior mathematics courses and the approach to programming that is used in CS1. In this respect, the study by Pioro (2006) took a step further by emphasising the importance of a particular type of mathematical background in relation to the students' performance in a CS1 course. According to the results students who had taken discrete mathematics and calculus courses before enrolling CS1 course performed better than students who had two calculus courses. Thus, the results implied that discrete mathematics course taught such problem solving techniques that were needed in solving programming problems.

In addition to academic experiences, students' perceptions of the learning environment, motivation, and cognitive skills have interested researchers. For example, student's comfort level and willingness to work hard were factors that arose in several studies. The study by Bergin and Reilly (2006) highlighted that in addition to mathematics grade there were two factors that predicted students' performance in an introductory programming course. Comfort level had a positive relationship with performance whereas game playing during the course had a negative effect. The study by Bergin and Reilly also suggested that weaker students had lower intrinsic motivation than stronger students. They also used fewer meta-cognitive strategies (planning, monitoring, regulating) compared to the stronger students. The study by Cantwell Wilson (2002) partly corroborated the results of the previous study. The results of Cantwell Willson's research suggested that students' comfort level and previously taken programming courses had a positive effect on students' success in CS1 level course. On the other hand, game playing (before the course) as well as the contribution of luck to success had a negative effect on the grade. The importance of the comfort level is further corroborated by the study by Ventura (2005). His results also suggested that the student's effort is also a strong predictor of success. The study by Bennedsen and Caspersen (2005) seconded that students who worked harder got better grades than students who worked less.

The notion that students who worked harder were more successful than students who worked less is not surprising. However, the decision to work hard is related to self-efficacy belief and thus the results are relevant. Self-efficacy has been studied in some papers; the study done by Ramalingam, La Belle and Wiedenbeck (2004) suggested that students' mental models and self-efficacy beliefs affected students' performance in the CS1 course. The study by Wiedenbeck (2005) further highlighted the role of perceived self-efficacy and knowledge organisation skills as factors that affected students' success in an introductory programming course. The results of the study by McKinney and Denton (2004) suggested that student's perceived competence, interest, effort, value, and lack of pressure were significantly correlated with the grade. However, the results also showed that in the CS1 course there was a negative, significant shift from pre-test to post-test suggesting that the students' perceptions on all previously mentioned factors were degraded. Relating to the self-efficacy theme the results of the study by Rountree,

Rountree, Robins and Hannah (2004) revealed that the expectation of a grade other than the best grade was associated with failure.

The research papers also suggested that student's cognitive skills are connected with the success. The study by Pillay and Jugoo (2005) highlighted that there was a positive correlation between the students' problem solving ability and programming performance. The study also revealed that students whose first language was not the same as that used in the course, performed less well than other students. The study by Mancy and Reid (2004) highlighted specific cognitive skill in learning programming. The results suggested that field-independent students who were better at restructuring the learning material in their own way, thus making sense of it achieved better marks in the examinations than did field-dependent students.

Finally, also student's learning skills have been discussed in relation to success and failure. The study by Bergin, Reilly and Traynor (2005) analysed the relationship between self-regulated learning and students' performance in an introductory programming course. The results suggested that students with high level of ability used more meta-cognitive strategies (planning, monitoring and regulating cognition) and resource management strategies (managing of time, effort, environment, and interaction with other students and teacher to seek help) than did students with low level of programming ability. The results also highlighted that students with high intrinsic motivation performed better than students with low intrinsic motivation. Also, students with high levels of task value also performed better than students with low levels of task value.

The study by Simon et al. (2006) highlighted that the surface approach to learning is negatively correlated with success. The study found also that students' spatial knowledge (measured by their navigational styles) is related to success. The paper also reported the qualities and skills students' perceived to be important in learning programming well. Among the three most often mentioned factors were logical thinking, problem solving, and attention to details.

Factors predicting success in degree level studies

Previous studies have discussed the success factors on a course level. Another branch of studies highlights the same topic but at the degree level. The study by Boyle et al. (2002) shed light on the relation between the university exit performance and a variety of variables, such as, prior academic achievements in high-school. The results may indicate that the level of prior mathematics qualification and prior computing qualifications did not relate with the success. Additional advanced level (A-level[4]) school examination did not influence university performance either. In addition, the university entry and exit scores did not correlate and there was no difference between traditional and non-traditional entrance. The study by Chamillard (2006) presented models that can be used to predict a student's overall performance in a particular course based on his/her performance in previous courses. The study suggested that the performance in some CS courses correlates with the performance in some other courses.

[4] After secondary school, students can start a two year programme, which leads to A (Advanced) level state examinations, which qualify them for entry into university. Students usually specialise in subjects that are relevant to the degree subject they wish to follow at university. To gain entry to a top UK university applicants need three or four good A Level grades.

The predictive models served also as a tool to identify key courses in a curriculum (key courses are the ones that provided the strongest predictive effect in the predictive models for later courses).

The study by Katz, Allbritton, Aronis, Wilson and Soffa (2006) focused on factors that predicted performance and persistence in an undergraduate CS program. The results suggested that persistence was highly correlated with achievement and many factors that predicted achievement predicted persistence. Among the factors that predicted persistence were math and verbal SAT scores, the number of calculus courses taken, prior computing experience, home access to a computer during high school, and having a role model during high school. In addition, factors such as interest in CS and positive attitude towards the CS department predicted achievement. The study also provided interesting results concerning why some high achieving students decided to leave the CS study program. The two main reasons were the loss of interest, which started with the loss of confidence. The study by Goold and Rimmer (2000) analysed various factors that affected performance in the first-year computing. The results highlighted the dynamic process of the timing and duration of the influences of the variables. For example, dislike of programming negatively influenced performance. In addition, its impact grew in second semester. The problem solving ability was found to influence the performance in two first courses to some degree but after that its role diminished. Finally, students' general academic ability was related to the performance of the computing units. That is, students' computing results conformed to the other undergraduate grades.

Cognitive, motivational, psychological, and social factors

The last few previous paragraphs have discussed the factors that predict student success in the CS1 course or in the CS degree in general. The following paragraphs concentrate on studies that do not aim at predicting success or failure but highlight factors that researchers have found to be problematic for students. Just as before, the literature review reveals various factors that may cause difficulties to students.

First, the studies by Lahtinen and Ahoniemi (2005) and Robins et al. (2003) highlighted students' difficulties with applying knowledge. Students may know the theory of programming, but when it comes the time to write actual programs it is difficult to apply the knowledge in practice. The study by Robins et al. continues by listing novice's surface and superficially organised knowledge, and the lack of detailed mental models as factors causing difficulties. Continuing with the cognitive factors the study by Vagianou (2006) introduced the concept of program working storage as a means to present difficult concepts and address novice programmers' pre-conceptions.

Second, the study by Forte and Guzdial (2005) highlighted the importance of motivation in learning programming. The paper addressed the differences in CS majors' and non-CS majors' motives to take CS1 course and introduced tailored CS1 courses for discrete groups of students. This measure was done to engage non-CS majors and enhance their motivation.

Third, the pedagogical actions are meant to help students to learn. However, in some cases unexpected negative effects may cover positive effects. The study by Kinnunen and Malmi (2005) reports the CS1 course for CS minors where problem-based learning (PBL) was used to enhance the learning outcomes and to lower the drop-out rate. The observations on the PBL group revealed that whereas some groups worked efficiently others suffered from severe group dynamic related difficulties.

Fourth, when students are studying programming they most likely face tasks that they find challenging. The paper of Simon, B. et al. (2008) recognised the differing ways students at CS1 course approached challenging tasks. The paper reported the authors' intervention that was aimed at affecting the students' fixed mindsets. The studies by Schwartzman (2006, 2007) too, analysed students' responses to difficult material in general and discussed how the teacher could engender students' reflective responses. The empirical data for the study was collected from an upper-level undergraduate course. However, the relevance of engendering students' reflective responses when they face difficult material is evident for teachers in CS1 courses, too.

Fifth, learning environment and social relations affect students. The study by Crenshaw, Chambers, Metcalf, and Thakker (2008) shed light on some aspects on student retention. The results highlighted that CS students felt isolated from each other, faculty, and greater computer science community. In addition, the studies of Barker and Garvin-Doxas (2004) Garvin-Doxas and Barker (2004) reported that the learning environment in computer science courses was defensive, which resulted in lower confidence among female students towards their major. The study by Foor, Walden and Trytten (2007) also discussed the learning environment and social relations related aspects of the students' experiences. The study shed light on implicit cultural aspects of the university. A case of multi-minority engineering student highlighted how the lack of social, cultural, and financial capitals affected the student, and resulted in a variety of difficulties.

As a summary, the difficulties that came up during the literature review are gathered in The difficulties are divided into categories. During the literature review, the emerging difficulties were listed. Later, the difficulties were divided into categories according to the source of the difficulty. This analysis resulted in seven categories that explained the source of the difficulty/challenge students may have faced: students' previous academic experience/previously acquired skills and knowledge, students' cognitive skills, students' psychological facets, learning skills, social/cultural aspects, teaching organisation (e.g., course arrangements), and other.

Table 3 has three categories: CS1 course level difficulty related factors, CS1 course level success related factors, and study module/degree level of the difficulty/success related factors. Success factors were included in the analysis as well because they have a potential to highlight the possible difficulties. The table can be read row by row. If the same aspect came up both as an aspect that causes difficulties and, related to the success, it was placed in both categories (i.e. in the same row in all the columns). For instance, poor mathematics skills were related to difficulties whereas good mathematics skills were related to success in both at the CS1 course level and at the degree level. Now because mathematical skills are found in the same row it is easier to discern its role in causing difficulties, contributing to success, and whether its role has any long-term significance (e.g., in a context of a degree level) according to the literature.

Table 3 Summary of factors relating to students' difficulties and success

Source of difficulty/success	CS1 course level		Study module/ degree level
	difficulty related factors	success related factors	difficulty/success related factors
Students' previous academic experience/ previously aquired skills or knowledge	- Poor math skills/poor math background (Beaubouef and Mason 2005)	- Mathematics skills (Byrne and Lyons 2001; Cantwell Wilson 2002; Bennedsen and Caspersen 2005; Bergin and Reilly 2006) - Discrete Mathematics course and calculus course (Pioro 2006)	- Math and verbal SAT scores (Katz et al. 2006) - The number of calculus courses taken (Katz et al. 2006)
		- Previously taken programming courses (Cantwell Wilson, 2002)	- Prior computing experience, Home access to a computer during high school (Katz et al. 2006)
		- Mental models (Ramalingam, LaBelle et al. 2004)	
	- Student first language is other than what is used in instruction (Pillay and Jugoo 2005)		
Student's cognitive skills	- Poor problem solving abilities (Beaubouef and Mason 2005)	- Problem solving ability (Pillay and Jugoo 2005; Simon, Fincher et al. 2006)	- The role of the problem solving ability diminished after the two first programming courses. (Goold and Rimmer 2000)
	- Cognitive skill: field-dependent (Mancy and Reid 2004)	- Cognitive skill: field-independent (Mancy and Reid 2004)	
	- Novice programmer's pre-conceptions (end-user stance) (Vagianou 2006)		
	- Students' preconceptions and knowledge about how demanding the course is (Meisalo, Sutinen et al. 2002)		
	- Ability to apply knowledge (Robins, Rountree et al. 2003; Lahtinen and Ahoniemi 2005)		
	- Knowledge structures :Surface and superficially organized knowledge, lack of detailed mental models (Robins, Rountree et al. 2003)		
		- Spatial visualization skills: landmark maps (Simon, Fincher et al. 2006)	
		- Logical thinking skills (Simon, Fincher et al. 2006)	
		- Paying attention to details (Simon, Fincher et al. 2006)	

Source of difficulty/success	CS1 course level		Study module/ degree level
	difficulty related factors	**success related factors**	**difficulty/success related factors**
Student's psychological facets	- Low motivation (Bergin and Reilly 2006; Forte and Guzdial May 2005)	- Intrinsic motivation (Bergin, Reilly et al. 2005)	
	- Low levels of task value (Bergin, Reilly et al. 2005)	-Perceived value of programming skill (McKinney and Denton 2004) - Interest (McKinney and Denton 2004)	- Loss of interest/ interest towards CS (Katz et al. 2006)
	- No confidence to pursue university-level studies (Xenos, Pierrakeas et al. 2002)	- Self-efficacy believes (McKinney and Denton 2004; Ramalingam et al. 2004; Wiedenbeck 2005)	- Loss of confidence (Katz et al. 2006)
	- Contribution to the luck (Cantwell Wilson 2002)	- Working hard, effort (McKinney and Denton 2004; Bennedsen and Caspersen 2005; Ventura 2005)	- Persistence (Katz et al. 2006)
	- Expectation of a grade other than the best grade (Rountree et al. 2004)		
	- (lack of) students' reflective responses (Schwartzman 2006; Schwartzman 2007)		
		- Comfort level (Cantwell Wilson 2002; Ventura 2005; Bergin and Reilly 2006)	
			- Dislike of programming (Goold and Rimmer 2000) - Positive attitude towards the CS department (Katz et al. 2006)
Learning skills	- Surface approach to learning (Simon, Fincher et al. 2006)	- Deep approach to learning (Simon, Fincher et al. 2006)	
	- Readiness and willingness to take the responsibility for own learning. (Sheard and Carbone 2007)	- Level of using meta-cognitive strategies (planning, monitoring, regulating) (Bergin, Reilly et al. 2005; Bergin and Reilly 2006)	
	- Too little time left for studies (Xenos, Pierrakeas et al. 2002)	- Using resource management strategies (managing of time, effort, environment, and interaction with other students and teacher to seek help) (Bergin, Reilly et al. 2005)	
	- Poor project management skills (Beaubouef and Mason 2005)		
	- Fixed mindset (Simon, Hanks et al. 2008)		
		- Knowledge organization skills (Wiedenbeck 2005)	

Source of difficulty/success	CS1 course level		Study module/ degree level
	difficulty related factors	success related factors	difficulty/success related factors
Social/ cultural aspects	- Defensive communication climate (Garvin-Doxas and Barker 2004)		- Defensive communication climate (Barker and Garvin-Doxas 2004)
	- Group dynamic (Kinnunen and Malmi 2005)		- Students feel isolated from each other, faculty, and the greater computer science community (Crenshaw, Chambers et al. 2008)
			- Lack of social and cultural capitals (Foor et al. 2007)
			- Having a role model during high school. (Katz et al. 2006)
Teaching organisation	- Tutor-teachers' negative attitudes (Meisalo, Sutinen et al. 2002)		- Poor advising before and during studies in college (Beaubouef and Mason 2005)
	- Lack of assistance from tutor-teachers (Xenos et al. 2002)		
	- Tutor-teachers' ability to tell about arrangements and the course itself (Meisalo, Sutinen et al. 2002)		
	- Graduate student teachers (Beaubouef and Mason 2005)		
	- Poorly designed CS1 lab courses (Beaubouef and Mason 2005)		
	- Lack of practice/feedback (Beaubouef and Mason 2005)		
		- Lack of pressure (McKinney and Denton 2004)	
Society			- Financial situation, need to work (Foor et al. 2007)
Other	- Playing games (Cantwell Wilson 2002; Bergin and Reilly 2006) - Heath-related reasons (Xenos et al. 2002) - Personal reasons (Xenos et al. 2002) - Family reasons (Xenos et al. 2002)		

2.3.4 Non-course content related issues that teachers find difficult to teach

In the earlier part of this chapter, I have listed course content related topics that teachers have found difficult to teach. The following paragraphs will concentrate on other types of difficulties that the literature review revealed. The difficulties that found varied from specific skills that were difficult to teach, to the alternating circumstances to prevailing student and to teaching cultures.

The study by Andersson and Bendix (2005) discussed the challenges that teachers face due to the rapidly changing requirements and proposed a set of pedagogical techniques that might help teachers. The (anecdotal) list of difficulties the authors recited included aspects, such as, the heterogeneous of students' qualifications and the background, decreasing resources for teaching per student, demand for new courses to satisfy the requests from industry, and the necessity to adapt to changes the study board's makes to the syllabus of the course on short notice. The authors suggested practices that allow shorter feedback loops to inform the teacher thus making it possible to react to changing situations. The study by Herrmann et al. (2003) partly seconded the previous list of changes teachers had to deal with. Authors reported three major reasons that have forced them to redesign their introductory programming courses. First, the computer science faculty has not grown at the same pace as the enrolment growth and thus the student teacher ratio is higher. Second, large-lecture hall instruction is not appropriate anymore to students whose computing experience and skill are widely divergent. Third, a rate of failure, withdrawal and D grades has varied by 25-50 percent.

The study by Waite, Jackson, Diwan, and Leonardi (2004) highlighted an important aspect by discussing the factors that may affect whether the chosen pedagogical means are successful or not. The paper discussed the facets of the CS student culture that hindered the effective usage of the group work. Aspects, such as, strong preference for working alone, a tendency to procrastination, "tinkering" –approach to assignments, seeing assignments as products rather than processes, combativeness, unwillingness to support others, and finally, the absence of passion were all facets of the culture that hindered effective group work. However, the authors noticed that the students were not alone responsible for their culture. There were instructional practices that fostered the culture described above. For example, assignments did not force the students to think about the process of learning, rather it was possible to complete them without giving any thought to the process. The explicit plagiarism polices enforced the students willingness to work alone and unwillingness to support others.

The study by Dale (2006) reported topics, which teachers found difficult to teach. One of the topics was student maturity. This category included a long list of student's skills: study skills, time management, need for hard work, thinking before coding, the importance of self-discipline, the importance of class attendance, good work habits, the need for students to believe in themselves, learning how to study, learning how to think, and planning.

The study by Sheard and Carbone (2007) analysed teachers' and students' perspectives on student-centred, technology-enhanced pedagogical program. The results revealed that there were not only differences in perspectives but also there were some difficulties the teachers faced in demanding settings. The results suggested that teachers showed a good understanding of student-centeredness but that they had difficulties with executing it in practice. On the other hand, first year students showed varying readiness and willingness to accept a student-centred approach that required them to take the responsibility for their own learning. The teachers also found that the nature of the degree program provided little possibilities for them to adapt curriculum and teaching to meet the needs of their students. Because the courses were taught over multiple campuses as well as in on-campus and distance education mode, the materials had to be well prepared before the semester began. In addition, the results also suggested that there were disparities in students' and teachers' perceptions on some other important aspects of instruction, too. First, the teachers emphasised in their lectures extrinsic sources of motivation (career in ICT) whereas students reported that they were intrinsically motivated. Second, teachers perceived the online learning environment as supplemental to classes whereas students perceived them as replacement for classes and thus they did not see the need to attend the classes.

The study by Carbone et al. (2007) shed light on computer science and IT teachers' conceptions of successful and unsuccessful teaching. Especially the conceptions of unsuccessful teaching revealed aspects that are challenging for teachers or are the sources of some specific difficulties. Five categories unfolded from the data: teacher lacked skills (e.g., organisation of material), teacher lacked organisational support (e.g., large class sizes, availability and skills of assistant teachers, high student-teacher ratios, inadequate time relief for preparation), student lacked responsibility (e.g., not working hard enough or not showing up), domain complexity, and student lack of understanding. The study by Tutty, Sheard and Avram (2008) corroborates the previous study by stating that the lack of support and encouragement from the university's side may resulted in teachers to adopt more teacher centred approaches to teaching against teachers' preferred teaching styles. Table 4 summarises other than course content related difficulties and challenges teachers confront.

Table 4 Other than course content related difficulties teachers confronts

Difficulty factors	CS1 course level difficulties	General difficulties
Students' previous academic experience/ previously acquired skills or knowledge	- Heterogeneous of students' qualifications and the background (Andersson and Bendix 2005) - Students' divergent computing experience and skills (Herrmann et al. 2003)	
Teaching cognitive skills	- Teaching how to think (Dale 2006)	
Student's psychological facets	- Teaching students to believe in themselves (Dale 2006) - Teaching the importance of self-discipline (Dale 2006) - Inaccurate perceptions on sources of students' motivation (Sheard and Carbone 2007)	- Student lacks responsibility (Carbone et al. 2007)
Teaching learning skills/ habits	- Teaching study skills: how to study, planning & good work habits, thinking before coding (Dale 2006) -Teaching time management skills (Dale 2006) - Teaching the need for hard work (Dale 2006) - Stressing the importance of class attendance (Dale 2006) - Students' readiness and willingness to accept student-centred approach (Sheard and Carbone 2007)	
Social/ cultural aspects		- Teaching culture (Waite et al. 2004) - Student culture: (Waite et al. 2004) -- Strong preference for working alone -- Tendency to procrastination -- "Tinkering" –approach to assignments -- Seeing assignments as products rather than processes -- Combativeness -- Unwillingness to support others -- Absence of passion
Course organisation/ teaching organisation	- Difficulties with operationalising the idea of student-centeredness in practice (Sheard and Carbone 2007) - Disparities between students and teachers how the role of the learning environment is seen (Sheard and Carbone 2007) - Pressure to react on poor learning outcomes: a high rate of failure, withdrawal and D grades (Herrmann et al. 2003) - Graduate student teachers (e.g., no training for teaching) (Beaubouef and Mason 2005) - Decreasing resources for teaching per student (Andersson and Bendix 2005; Herrmann et al. 2003) - The nature of the degree program provides little possibilities for teacher to adapt curriculum and teaching to meet the needs of their students. (Sheard and Carbone 2007) - Poorly designed CS1 lab courses (Beaubouef and Mason 2005) - Lack of practice/feedback in CS1 course (Beaubouef and Mason 2005)	- Changes the study board's makes to the syllabus of the course in short notice (Andersson and Bendix 2005) - Poor student advising before and during studies in college (Beaubouef and Mason 2005) - Large class sizes, high student teacher ratios (Carbone et al. 2007) - Availability and skills of teaching assistants (Carbone et al. 2007) - Inadequate time relief for preparation (Carbone et al. 2007)
Teacher	- Teacher lacks skills (Carbone et al. 2007)	
Society		- Demand for new courses to satisfy the requests from industry (Andersson and Bendix 2005)

2.4 Summary of the literature review

The literature review revealed several different types of challenges and difficulties students and teachers had confronted during the instructional process. The content of the introductory programming course was a source of difficulties that several researchers had studied. Many studies analysed which course topics were difficult for the students to learn. However, which topics were difficult for the teacher to teach was a less studied area.

Already du Boulay (1986) listed five areas of difficulty that a novice programmer may confront: problem of orientation, notional machine, notation of the formal language, structures, and pragmatics of programming. The literature review revealed that also studies that are more recent found these areas problematic. For example, arrays, recursion and program design were topics that several studies found difficult for the students to learn.

In addition to the course content related difficulties, the literature review revealed several other than programming language related topics that may affect the students' performance in CS1 course or in CS degree in general. Based on the analysis of the sources of students' difficulties divided into seven subcategories: students' previous academic experience, students' cognitive skills, psychological facets, learning skills, social and cultural aspects, teaching organisation, and others. Based on the literature review it seems that the mathematics background is in relation to the student's performance in CS1 course and in CS degree in general. Poor mathematics skills are related with difficulties and correspondingly strong mathematics background is related with the success both in CS1 course and in degree level. However, not all studies seconded the importance of the mathematics background.

Problem solving skills proved to be another type of skills that were related to the difficulties and success. Poor problem solving skills caused difficulties whereas better skills were related to success in CS1 course. Interestingly, the role of problem solving ability seemed to diminish after a couple of first computer science courses. Motivation, interest, perceived value of programming skill, self-efficacy believes and persistence were the psychological factors that correlated with difficulties and success. In addition, interest self-efficacy/confidence, and persistence were factors that contributed to the success in a degree level, too.

From learning skills, poor time managing skill was directly related to dropping out of the CS1 course whereas resource management skills, and time management skills as a part of learning skills, was paired with success. Deep and surface approach to learning was also related to the students' performance. From the social and cultural factors the defensive climate as well as the feelings of being isolated from peers and faculty was related with the difficulties. In addition, the students' lack of social and cultural capitals proved to cause some difficulties. The course arrangement was also mentioned as one possible source for difficulties. Poorly designed courses, graduate student teachers, and tutor-teachers' negative attitudes were seen as possible factors relating to the difficulties.

The literature review revealed also other than course topic related factors that may cause teachers difficulties or at least they put a stress on the teacher to act on a certain way. The sources of difficulties were divided into seven subcategories according to the theses that emerged as a result of the analysis: students' previous academic experience/previously acquired skills or knowledge, teaching cognitive skills, student's psychological facets, teaching learning skills/habits, social/cultural aspects, course organisation/teaching organisation, society.

For example, teachers found it difficult to teach students study and time managing skills in addition to make students understand the importance of hard work and self-discipline. The heterogeneous of students' experience and skills was also challenging for teachers. In addition, the teaching organisation was a source of difficulties and pressure teachers' had to face. For example, the increasing student teacher ration and the nature of the degree program that left little room for teacher to react to the students' difficulties did not leave teachers with many options. The studying and teaching cultures were also found as factors that made it difficult to apply some pedagogical approaches efficiently. Finally, the demands of the industry concerning new courses affected teachers.

The closer analysis of Table 1 - Table 4 brings out the similarities and disparities in factors that cause difficulties for students and teachers. For example, students' previous academic experience came up as a factor that may cause difficulties. From students' point of view, it meant that poor math skills might make it harder to succeed in CS1 course. From teachers' point of view, the heterogeneous of the students skills and background was challenging. Therefore, even if the name of the subcategory of the difficulties is roughly the same in Table 3 and Table 4 the content of those subcategories have differences.

Figure 5 illustrates the sources of difficulties or challenges that emerged from the literature review. For example, the analysis revealed several course topics that students had difficulties with. At the same time, the literature review suggested that some of the course content were difficult for the teacher to teach as well. Learning skills and teaching organisations were the topics that caused challenges both for students and teacher. Poor learning skills complicated students learning and teachers felt that studying and learning skills were difficult to teach. Poorly designed courses and high student teacher ratio as the teaching organisation level factors made the teaching and learning process harder. From the students' perspective, the student's psychological facets and cognitive skills were a source of difficulties, too. On the contrary, the literature review suggested that students' cognitive skills were not causing great difficulties for the teacher. In general, the literature review highlighted several student related factors that were causing difficulties for students. Likewise, the same characteristics were the source for teachers' difficulties, but the literature review revealed fewer factors from that viewpoint.

Figure 5 gives an idea of the similarities and disparities of the challenges the students and teachers in CS1 course may face. Due to the limited focus of the literature review, the papers that were selected to this analysis discussed the learning and teaching of computer science. However, many difficulties and challenges that emerged as result of the literature review have been widely discussed in a field of engineering education and in higher education: self-efficacy (e.g., Valentine et al. 2004), students approaches to studies (e.g., Jungert 2008), and attrition/retention (e.g., Baillie and Fitzgerald 2000; Tinto 1975). The results of those studies are not included into the literature review since the goal of the review was to shed light on the variety of the challenges reported in the field of computing and computer science education.

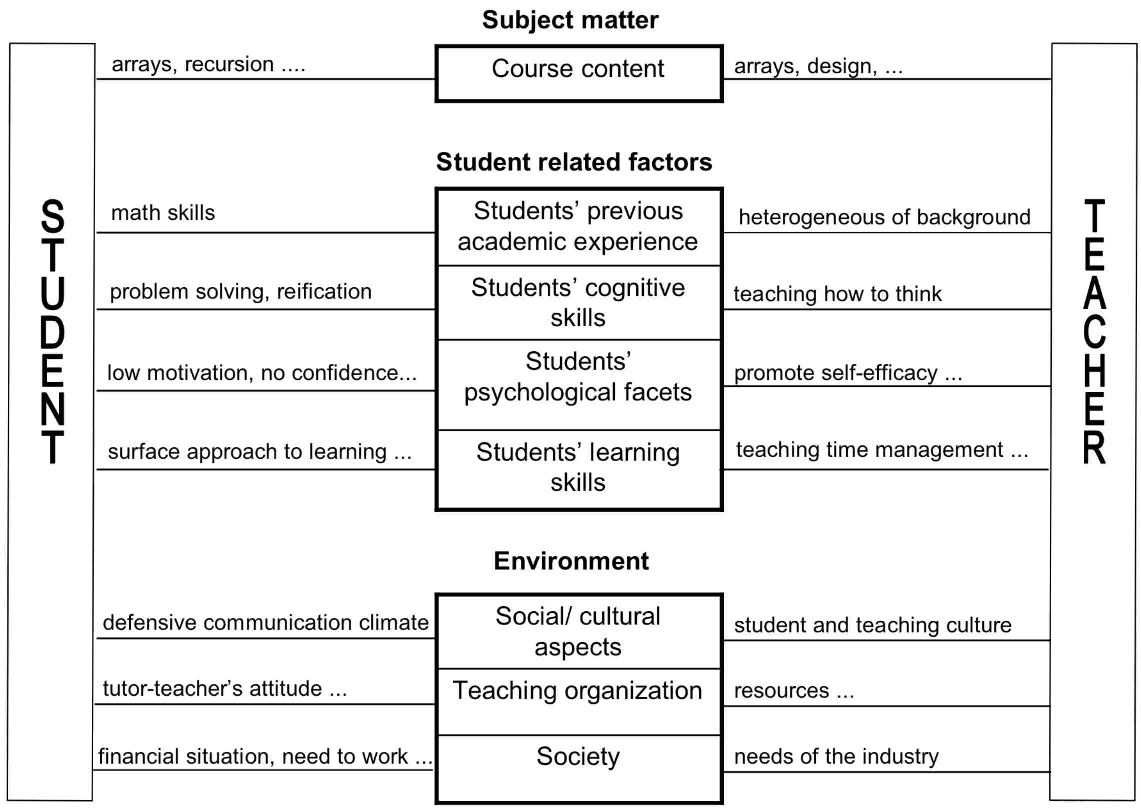


Figure 5 Factors causing challenges for students and teachers

The literature review highlighted briefly the importance of motivation in learning programming. Over the years, motivation has been studied from many viewpoints and as a result there are several theories that try to explain the nature and the source of motivation. For instance, goal theory (see e.g. Covington 2000) suggests that there is a close relation between the different types of goals and the types of motivation. Achievement motivation theory (see e.g. Elliot and Church 1997) highlights the importance of how much a person values the outcome. Social cognitive theory (see e.g. Bandura 1997) emphasise the role of expectations of success and self-efficacy beliefs. The relation between students' goals and motivation is further discussed in chapter five. However, this research does not concentrate especially on the role of motivation in student's studying process but aims at observing other influencing factors, too.

Further analysis of the studies included into this literature review highlights that some areas are much more researched than others. For example, there were only few studies that analysed the teaching organisation level factors. The few studies that discussed the teaching organisation mostly based the discussion on anecdotal knowledge. These papers were included into the analysis to highlight the variety of challenges and to draw the attention to that this is a field that needs further research.

An aspect that was missing in the literature review was the process nature of the instructional process. The vast majority of the papers analysed "snapshots" of the instructional process and not the process of studying and teaching. Thus, one central characteristics of instruction has been overlooked. Finally, another aspect that was largely missing in the literature was the research that would view the instruction from different points of view. There were some studies that had surveyed both students' and teachers' conceptions. However, they were a minority. This might be due to the restrictions of publication forums. Especially, the conference papers have such a restricted page limitations that it may have forced the authors to choose a very limited focus. Nevertheless the reasons for restricted focuses, the community of researchers and teachers would benefit from more comprehensive and holistic approaches that would shed light on the instructional process from several viewpoints.

3 Research questions

This chapter describes the research questions, which evolved during the research process. What started out as a rather straightforward question concerning students' reasons for dropping out of a certain course, turned into several interesting questions considering not only students but also teachers and the teaching organisation. The finalised research questions are a result of a process that took place in parallel with the actual data collection and analysis process. A closer observation of the drop-out phenomenon revealed that it is rather complex. Dropping out does not happen in isolation – it is affected by several factors. The literature review revealed several different kinds of difficulties and challenges students might face during the instructional process. A few studies shed light on the students' reasons for dropping out of courses. However, most of these studies presented only a snapshot of the situation, and did not regard the process nature of instruction. From this notion emerged the need to look at the drop-out phenomenon as an indication of a malfunctioning instructional process. In the end, the main focus had shifted from discovering the reasons for dropping out of an introductory programming course to understanding the instructional process better. The basis of this shift is justified: Limiting the focus of the study on one undesirable outcome may restrict our understanding of the complexity of the phenomenon. More general focus makes it possible to understand and analyse various outcomes and reasons behind them.

As the literature review suggested, students had difficulties that originated from the subject matter, students' characteristics and the environment. These difficulties concerned both the students and the teachers. It was also evident that some of the challenges depended on the actions of the teaching organisation. From such observations, I derived three research questions in order to shed light on the instructional process from three different points of view: the students', the teachers' and the teaching organisation's. On a practical level, the aim of this research is twofold. On the one hand, this research is looking for the reasons for a malfunctioning instructional process, and on the other hand, it aims at highlighting aspects of the process that need further development.

Students' point of view

The initial motive for this research project was a concrete problem in an introductory programming course (CS1) at Helsinki University of Technology (TKK). At the time this project started in the spring 2005, around 40% of the students who enrolled in the course dropped out. The first step in planning the interventions was to find out what kind of difficulties and challenges students faced in these courses. Thus, the first set of research questions is:

1. What kind of difficulties do students encounter when studying computer programming?

1.1. Why do students decide to drop out of the CS1 course?

1.2. Which content-related issues do students find difficult?

1.3. Are there statistically significant differences in what programming related issues students who drop out and students who pass the course find difficult?

1.4. What kind of strategies to overcome difficult issues do students find helpful?

As soon as the first results started to emerge, it became evident that the reasons for dropping out are manifold and they relate to the students, the course (content, teaching personnel etc.) and even to the university as an organisation. As a result of this observation, it became clear that I needed multiple points of view to understand the reasons behind the high drop-out rate. Adding the teachers' and the organisation's points of view to the study considerably broadened the focus of the research project, while steering the analysis to a more general level. For example, the focus shifted from reasons for dropping out to the kind of difficulties the students face during the course. Thus, the focus extended to all students in the course and their point of view of the course instead of just focusing on the students who dropped out. Furthermore, including the teachers' and the organisation's points of view to the study made it possible to analyse the course as a part of a larger system. The research procedure relating to the first research question and its sub-questions is described in chapter six (section 6.1) and its results in chapter seven.

Teachers' point of view

The course personnel, especially the teachers, are in a central role when it comes to planning and implementing interventions to help students with their difficulties. This research contributes to the existing literature by first focusing on how computer science teachers see their students' studying process. The focus is not on teaching and it is not on learning outcomes, but on the students' studying process and the teacher's possibilities of affecting it. Studying is defined as the actions that the student does to achieve (course) goals. Learning is distinct from studying and is considered as a desired outcome of the studying process. Teachers' perceptions of their students' action, studying, and teachers' thoughts what they can do to enhance studying adds new aspects to the current literature. The research questions relating to the teachers are:

2. **How do computer science teachers see the instructional process?**
   - 2.1. How do computer science teachers define what studying is?
   - 2.2. Which aspects of studying do computer science teachers think students have difficulties with?
   - 2.3. Which aspects of studying do computer science teachers think are focal for successful studying?
   - 2.4. How do computer science teachers think that they can affect the students' studying process?
   - 2.5. How do the computer science teachers consider the different phases of the instructional process?

The research procedure relating to the second research question and its sub-questions is described in chapter six (section 6.2) and its results are presented in chapter eight.

The teaching organisation's point of view

The third research question targets the teaching organisation's point of view in the instructional process. The students' and teachers' instructional processes that are analysed in this research take place in a university of technology. Thus, the students' and teachers' experiences and perceptions are integrated into the larger university system.

Clark (1979) suggests that there are four different manners of academic coordination: bureaucracy, profession, politics and the market. This division also provides different possible points of view from which the university could be approached. One possibility to approach the university is to see it as an organisation that provides the administration and financial resources to realise the university's pedagogical and research goals. That is, one could concentrate on the bureaucracy and to some degree on the politics side of the university. However, taken the original research question that related to students' difficulties during studying, the point of view that helps to understand instructional processes is the one that focuses on profession, especially on the university's actions to educate new professionals. Therefore, the university's actions that relate to the content and quality of the instructional processes are highlighted in this part of the study.

With the professional or pedagogical point of view in mind, the question emerges - what can the university's organisation do to enhance teaching, studying and learning? There are several aspects that affect instructional processes, yet are too large or not practical to be tackled by a single teacher. In those situations, organisational procedures and committees are called for. The focus of this part of the study was twofold: first analysing how the formal documents steer the students' and teachers' instructional processes, second, studying various aspects that play a part in administrations' tasks. The research questions relating to the organisation's point of view are:

3. How is the instructional process seen at the organisation level?

3.1. How are the different phases of the instructional process and the feedback about the process and outcomes presented in documents?

3.2. How do representatives of the administration see the instructional process?

The research procedure relating to the third research question and its sub-questions is described in chapter six (section 6.3) and its results are presented in chapter nine.

4 The analysis models: The "dimension doughnut" and the three-layered didactic triangle

This chapter introduces two analysis models that were developed to tackle the research questions that where introduced in chapter three. The models are the "*dimension doughnut*" and the *three-layered didactic triangle*. In the first part of this chapter, the multiple dimensions of the instructional process are described with the help of the developed "dimension doughnut" model. The "dimension doughnut" is one way to visualise the complex educational reality. Even though the two-dimensional representation cannot do justice to the complexity of educational reality, the "dimension doughnut" is a way to highlight some of the diverse viewpoints from which the instructional process can be analysed. The limitations and strengths of the "dimension doughnut" are also discussed in this chapter.

The second part this chapter, focuses on different kinds of interactions during the instructional process. The didactic triangle is taken as a starting point for this discussion. However, the original didactic triangle (see e.g. Peterssen 1989) was developed so that it can be used to analyse instructional processes not only on a course level, but also at the organisational level as well a societal level. Finally, the three-layered didactic triangle is used as a base for a meta-analysis of educational research. Eight meta-level categories of research are derived from the triangles, and the studies that were introduced in the literature review in chapter two are placed in these categories. This categorisation sheds light on the didactic focus of the research in the fields of computing education (CE) computer science education (CSE) thus highlighting an aspect of research that has not been considered earlier in this field of study. In conclusion, the limitations and potential of the didactic-focus-based categorisation system are discussed along with observations concerning research in the field of CSE.

4.1 The development of the "dimension doughnut"

Instructional process is a complex phenomenon that can be seen from several points of view. This chapter proposes a dimension-based approach to manage the complexity. One way to deal with the complexity of the instructional process is to think of the process as having several dimensions. This dimensional approach can be applied to examine a particular aspect of the instructional process. For example, Carbone (2007) developed a typology for describing tasks. She came up with eight dimensions that comprised a task typology, which can be used to characterise tasks (e.g., programming tasks). The dimensions were: routine-novel, closed-open, artificial-authentic, degree of ownership, degree of linkage, degree of reflection, individual-collaborative, and simple-complex. For instance, the routine–novel dimension refers to the novelty of the activity (not the novelty of the content) and the open-closed dimension refers to whether the students have a choice concerning the way the task could be tackled or what sort of product they will produce. These dimensions provide a way of thinking about strengths and limitations of tasks and thus provide teachers a valuable tool that can be used when planning, e.g., programming assignments. Carbone's task typology provides a tool to analyse one aspect of the instructional process in a detailed manner. However, this particular tool is not useful when discussing other aspects of the instructional process.

The dimensional approach can be applied to gain a better understanding of larger phenomena, too. For example, when a teacher looks at the instructional process as a whole, students' view of the instructional process is one dimension, teachers' view is another, and the content of the course provides a third dimension. Further, the study by Meisalo (1985) introduced six dimensions in a context of science education that can be used, for instance, to analyse pedagogical methods (the dimensions are also represented in a context of Finnish computer science education in the book by Meisalo, Sutinen and Tarhio (2003, p. 51)). The dimensions are: the subject matter that is taught, the nature of the didactic process, the nature of the interface with the computer, the connection to nature (e.g., the level of concreteness, the level of the cognitive process, and the nature of interaction (Figure 6). In the following paragraphs, I have taken some of the previously listed dimensions and some new ones under closer consideration. The new dimensions that I included into this analysis are the instructional entity and the study activity. I will discuss each dimension in detail later.

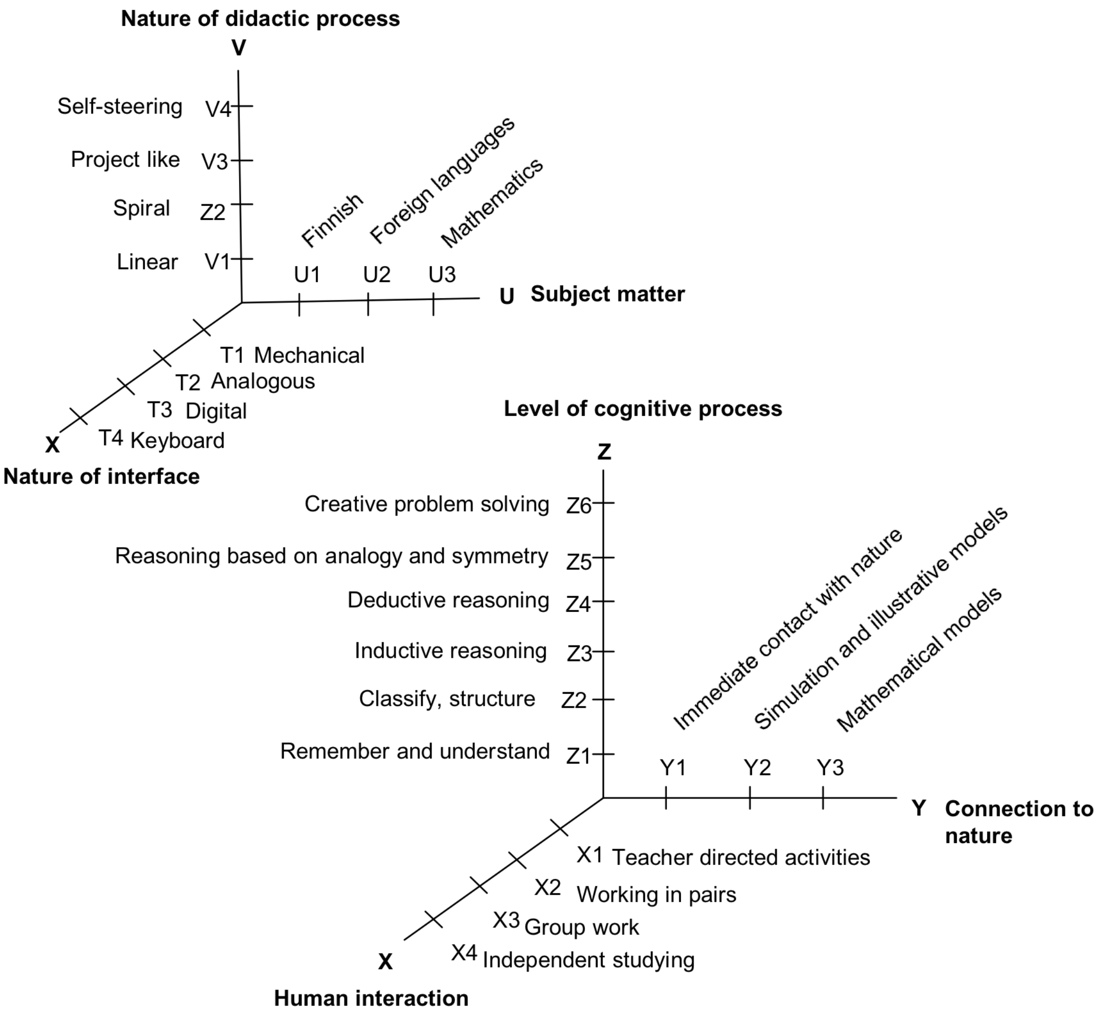


Figure 6 Six educational dimensions (Meisalo 1985)

The different dimensions of the instructional process can be seen as a multi-dimensional space where all the dimensions are visualised in the same picture. In Figure 7 I have visualised some dimensions that shed light on the instructional process: dimension A stands for students' point of view, B for teaching organisation, C for content, D for instructional entity, E for level of cognitive process, F for didactic activity, G for level of concretising, H for study activity and I for interaction (the colour coding will be explained later). The dimension "doughnut" aims at highlighting in a visual form that there are several dimensions simultaneously present in an instructional process. The figure can be used also by highlighting only two or three dimensions at a time, thus viewing sections of the multi-dimensional space.

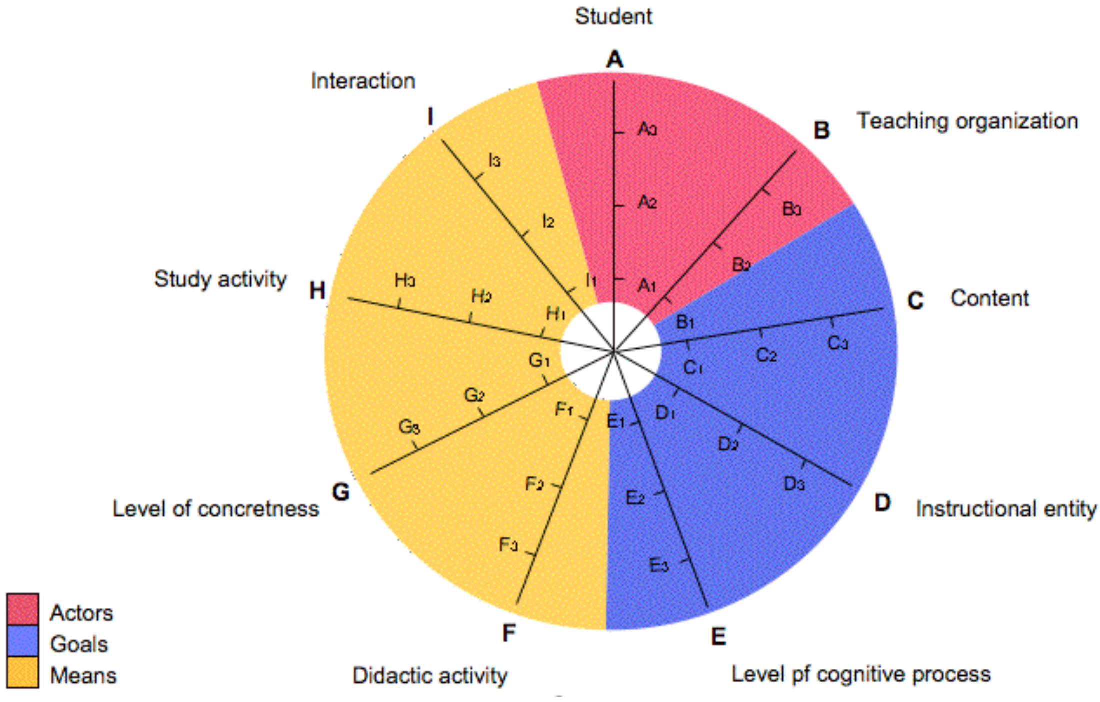


Figure 7 A "dimension doughnut"

Before I elaborate on each dimension, a few things need to be clarified. First, the description and Figure 7 are simplified abstractions, which cannot fully capture the richness of reality. There could be more than those nine dimensions and each dimension may have more than three categories that are now visualised in Figure 7. Second, the dimensions presented are non-orthogonal, which means that many of them may share aspects with other dimensions in the "doughnut". The aim here is not to find an orthogonal representation of the dimensions of the instructional process but one that serves as a viable framework in this study's context. Thus, the chosen dimensions may well contain aspects of other dimensions.

A practical consequence of complex pedagogical reality, which leads to a model with multiple dimensions and categories, which in turn leads to that when doing research on the instructional process there are a vast number of viewpoints to choose from. For instance, one could look at the instructional process from the point of view of a small group of students on a single course studying and learning a particular concept. On the other hand, one might review the instructional process from a curricular point of view. The selection of dimensions and categories depends on the researcher's point of view.

The nine dimensions that I have chosen can be divided into three sectors: *Actors* (rosy background), *Goal* (blue background) and *Means* of reaching the goal (yellow background). The sector *Actors* represents the teaching organisation, and consists of dimension *A Students* and dimension *B Teachers*. The next sector is *Goal* and consists of the dimensions *Content (C), Instructional entity (D),* and *Level of cognitive process (E)*. These are the identified components of actors and goal, which set the content for dimensions of *Means* in several ways. The last and most extensive sector is *Means*, which consists of the dimensions *Didactic activity (F), Level of concretness (G), Study activity (H)* and *Interaction (I)*. The common factor of these dimensions is the concrete activity of teaching or studying. Each of these dimensions highlights different aspects that the teachers and the students confront during the actual instructional process: students as they strive to learn and teachers as they strive to scaffold students' learning.

The number and the content of the dimensions and the way they are divided into sectors is a subjective issue, which varies according to the use of the framework. The division presented here is customised to fit the needs of this research. It highlights three important aspects of the instructional process: the actors, the goals, and the means teachers and students use to reach their goals.

Actors: the student and the teaching organisation

Dimension A concentrates on the student. This dimension contains categories, such as, *single student* (A_1), *group of students* (A_2), or, for example, *all the students' of the particular university* (A_3). The student dimension thus highlights the variety of viewpoints that emerge just by varying the number of students that are investigated. Dimension A does not have a counterpart in Figure 6. It was essential to include this new dimension in this study as the way students perceive the instructional process is one of the three points of view that are discussed in this study.

In the same way dimension B*, Teaching organisation* contains categories, such as *individual teacher* (B_1), *a group or team of teachers* that are responsible for certain courses or a larger module (B_2) and *all teachers and administrative staff of the university as representative of a larger teaching organisation* (B_3). Therefore, the categories of dimension B offer different viewpoints in the same way as the categories of dimension A. However, dimension B is more complex than dimension A. Due to their profession, teachers are part of a larger educational organisation, which sets ground rules for the teachers (e.g., curricula and strategies). Individual teachers and teams of teachers are often closely connected to different levels of organisation. For example, an individual teacher at a university might be solely responsible for a single course. A team of teachers and assistant teachers could be responsible for a set of courses. A department is responsible for the teaching of a particular major as a part of a degree programme. Faculty as a whole is responsible for teaching within a particular discipline. The board of directors of the university takes responsibility for the entirety of teaching at the university.

At Helsinki University of Technology (TKK), to take a specific example, a teacher together with several assistant teachers is responsible for a CS1 course, which is offered to CS majors as well as minors. The Department of Computer Science and Engineering takes responsibility for education in computer science. It also offers basic education in computer science to all students at the university. The Faculty of Information and Natural Sciences is responsible for producing professionals in computer science and information technology. The board of directors is the highest decision-making body at TKK. Ultimately, it is responsible for the entirety of educational activities at the university. Finally, the Ministry of Education sets guidelines for TKK as a part of the

Finnish higher education, for example, by granting money according to the number of annual graduates.

Dimension B does not have a counterpart in Figure 6. The inclusion of this new dimension was justified because the teachers' and the teaching organisation's points of view are two of the three points of view that are discussed in this study.

Goal: content, instructional entity and level of cognitive process

The *Goal sector* highlights a variety of issues that affect the goal-setting process. Goals refer here to the aims and expected learning outcomes that students and teachers/teaching organisations set for the instructional process. Dimension C, *Content*, portrays a central aspect of goal setting. Content can be divided, for example, into different subject areas, such as, *mathematics* (C_1)*, physics* (C_2) *and computer science* (C_3). Furthermore, each subject area can be divided into smaller content areas. As the analysis reaches yet smaller entities, different content constituents can be seen on a course level. For example, in an introductory computer programming course (CS1), sub-areas could include specific topics, such as basic loop structures, and more general issues, such as the fundamentals of object-oriented programming. Dimension C is similar with the dimension U *Subject matter* in Figure 6.

Dimension D, *Instructional entity*, refers to the scale or extent of the object of the instructional process. When discussing studying, one can concentrate on small entities, such as, *single concepts* (D_1)*, larger entities of knowledge or skills in a course or set of courses* (D_2), or even *curriculum-level entities of knowledge and skills* (D_3). The extent of the object of study is naturally related to a time scale. For instance, there might be specific *short-term goals* (D_1) that state on a concrete level what the student is expected to be able to do after a single course or lesson. There are also *long-term goals* (D_2) that cannot be reached after a single course, but require years of study.

The role of the instructional entity comes up when teachers are planning the instructional processes. For example, an individual teacher might focus on smaller entities, which are relevant to his/her course. The content and the goals must be aligned with the time reserved for the course. On the other hand, the team of teachers that is responsible for a set of courses needs to focus on larger entities in order to form an effective and coherent whole.

Dimension D does not have a counterpart in Figure 6. This dimension was included into the model to highlight the variation in the extent of the objectives of the instructional process. This variation brings yet more viewpoints to the instructional process.

The last Dimension E in the *Goal sector* is *Level of cognitive processes* (dimension E), which can contain processes from *remembering and understanding* (E_1) to *creative problem solving* (E_3). In other words, dimension E deals with the profundity of learners' cognitive processes. As the teacher sets goals, one of the issues that must be decided upon is the level of cognitive process that is targeted. The dimension of cognitive processes also has a close relationship to context. Different contexts might require different cognitive processes, and some processes are prerequisites for others. For example, in an introductory computer programming course (CS1), the nature of the subject directs the nature of the goals. A strong emphasis on programming skills in addition to conceptual knowledge is one instance of the close relation between the goal of the course and the nature of the subject. Dimension E equates the dimension Z *Level of cognitive process* in Figure 6.

Means: didactic activity, study activity, interaction and level of concretness

The last sector contains dimensions that describe the different means that the teachers and the students use in order to reach their goals. *Didactic activity* (dimension F) refers to different ways of teaching. There are many possible pedagogical approaches a teacher can choose from, such as *PBL* (F_1)*, learning through projects* (F_2)*, and lecture based teaching* (F_3). Dimension F thus highlights the practical pedagogical choices the teacher has when planning and teaching. The different ways of teaching are the teachers' concrete pedagogical means to scaffold learning. Dimension F equates closely the dimension V the *Nature of the didactic process* in Figure 6 - there is only a subtle difference in the emphasis. *Didactic activity* (dimension F) focuses more on the teachers' actions in choosing the didactic activity whereas *Nature of the didactic process* (dimension V in Figure 6) focuses more on the nature of the process itself.

Dimension G, the *Level of concretising*, addresses the level of how closely instruction is connected to real life. The different categories in this dimension reflect the abstraction level of the instructional entity. For example, G_1 reflects *the instructional event or process that is strongly connected to the concrete experiences* and thus the abstraction level is low. G_3, on the other hand, represents an instructional *event or process that discusses the subject matter on a high abstract level*. Dimension G's counterpart in Figure 6 is dimension Y *Connection to nature*.

Study activities (dimension H) are the students' concrete means to study. There is a vast number of ways to study. For instance, a student can decide to *read given study material, such as books* (H_1)*, do exercises* (H_2)*, or discuss the subject area with other students* (H_3). The combination of study activities that are used in a particular course naturally varies according to the content of the course, instruction, resources, such as, time, and personal preferences. Dimension H does not have a counterpart in the dimensions represented in Figure 6. This new dimension was included into the model to introduce the students' actions. In a sense dimension H (*Study activities*) is dimension F's (*Didactic activity)* pair. The former highlights students' activities and the latter teachers' activities.

Dimension I, *Interaction*, includes different types of interaction between the student, the teacher and the environment. For instance, instructional processes could emphasise *the student-student* interaction through small group activities (I_1), *interactions with the teacher* (I_2), or *interactions with the environment* (I_3)*,* such as books or e-learning environments. Dimension I equates the Dimension X *Human interaction* in Figure 6. The difference is that Dimension I includes interaction with the environment in addition to interaction with people. Dimension X *Human interaction* in Figure 6 concerns only human interaction.

In summary, dimensions A (*Student*), B (*Teaching organisation*), D (*Instructional entity*), characterise elements that can be discussed in different scales of size of analysis entity: individual – a group of people/organisation, single concept – curriculum. Dimensions E (Level of cognitive processes) and G (Level of concretness), on the other hand, contain categories that can be put into an order of some kind, such as, less abstract – more abstract. For example, creative problem solving could be valued as more demanding cognitive task than merely being able to remember and repeat. Dimensions C (*Content*), F (*Didactic activity*), H (*Study activity*), and I (*Interaction*) are not scalable.

In the previous paragraphs, I have described the multi-dimensional nature of the instructional process. Nine dimensions were identified as examples of dimensions that illuminate the instructional process as a phenomenon. This study aims to identify challenges students and teachers and teaching organisations face during the instructional process. Thus, in this study the dimensions A and B (actors) are in primary focus. However, since this study has a holistic approach many of the other dimensions, such as didactic activity and study activity are also discussed.

4.2 The roots of the didactic triangle

The “dimension doughnut” introduced in the first section of this chapter describes different points of view from which the instructional process can be seen. However, the “dimension doughnut” does not bring out the interaction between the different dimensions nor the process nature of teaching and learning. This section concentrates on another analytical model, the *didactic triangle,* which emphasises interaction during the instructional process. The didactic triangle is further developed and used as a base for analysing research in the field of computer science education.

The origin of the didactic triangle (Figure 8) goes back to the beginning of the 19th century to the works of Johann Friedrich Herbart (see e.g. Peterssen 1989). He introduced the triad of learner-teacher-content and emphasised that the relation between *teacher* and *learner* is not direct but that *content* stands between them. The didactic triangle describes these three main elements of a didactic system and the interrelations between them. The arrows in the triangle stand for relations between the three elements as they appear in the institutional instructional process. Arrow A stands for the teacher’s relation to the content. Arrow B stands for the teacher’s relation to or conceptions of the student and vice versa. Arrow C stands for the relation between the student and the content, which expresses itself, for example, as studying (Kansanen 2003).

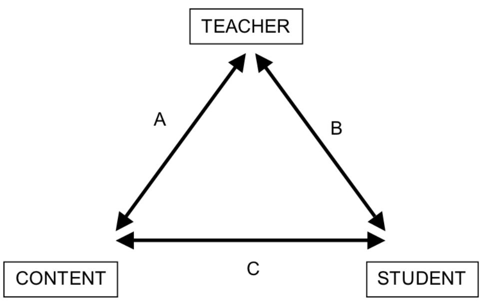


Figure 8 The didactic triangle (see e.g. Peterssen 1989)

The didactic triangle is a schematic representation (model) that enables the analysis of its components and the multi-level relations between the components. The representation is on a high abstraction level, which enables several variants of meaning concerning the meaning of the arrows and thus the model serves as a tool for analysing several aspects of instruction. Künzli (2000) argues that, depending on the emphasis, each strand (or arrow) in the triangle can have more than one meaning. For instance, arrow C in Figure 8 can be interpreted so that abstract phenomena are rated high or so that informal aspects of learning are highlighted. Furthermore, the relationships in the didactic triangle stand not only for concrete practice but also for theoretical cognitions. Friesen (2006, p. 50) summarises the relationships between the elements: ‘*all of the relationships in the Didaktik triangle should also be understood ... in terms of a* *"comprehensive intertwining of action and reflection, practice and theory" – an understanding of theory and practice in which the relationship between the two is an object of explicit concern and reflection.*

Bergamin (2006) brings out further evidence that the didactic triangle as a theoretical tool is still "alive" in the sense that it is found useful and is being developed further. The study by Bergamin developed the triangle by adding *community* to the triangle, creating a tetrahedron. In addition, it analysed the elements of teaching and learning situations in the context of blended learning[5] and stated that community is one essential, fundamental characteristic of co-operative learning. In Bergamin's model, the community is an equal node in the tetrahedron along with teacher, student and context. Kansanen and Meri (1999) and Kansanen (2003) have also developed the triangle further by adding another arrow (D), as shown in Figure 9. This arrow stands for the didactic relation between the teacher and the students' studying and learning processes. This didactic relation can be externalised, for example, by giving lectures or providing a learning environment.

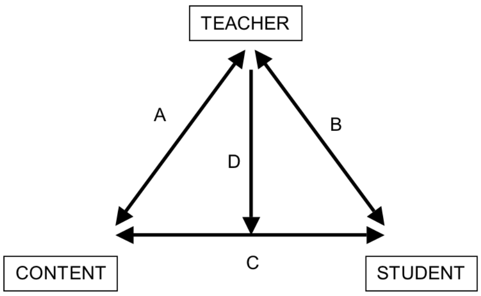


Figure 9 The didactic triangle as presented by Kansanen (2003).

4.3 Development of the three-layered didactic triangle

In the following, the elements of the didactic triangle as presented by Kansanen (2003) and the interrelations between them are taken as a starting point for further development of the triangle. The triangle presented by Kansanen is taken as a starting point since it brings into the foreground the didactic relation between the teacher and the students' studying and learning processes. I perceive this relation as an important aspect of the instructional process that other versions of the didactic triangles failed to emphasise. The development of the triangle has a concrete goal: to use the developed triangle as a starting point for a didactic focus based categorisation of educational research. This new categorisation system aims to highlight the didactic foci of studies in the field of computing education (CE) and computer science education (CSE). It thus serves as a tool to analyse which aspects of the instructional process are less studied and thus it has the potential to discover new and relevant research questions.

As an example of how the new categorisation system can be used an analysis of the didactic focus of research is executed on a delimited set of research. The analysis gives an example of how the didactic focus of research in the fields of CE and CSE, which has not been highlighted earlier, can be emphasised. The new categorisation system also

[5] Blended Learning is a blending of different learning methods, techniques and resources. For instance, it is learning, which combines online and face-to-face approaches.

opens new opportunities for determining which areas of research have been thus far overlooked.

The triangle presented in Figure 9 (Kansanen 2003) visualises the interaction between the three elements (student, teacher, and content) and offers a base for categorisation. However, the triangle in Figure 9 does not describe the richness of the reality in which instructional processes exist. The three main elements and their interrelations are inadequate as a base for categorisation of educational studies in computing education (CE) and computer science education (CSE). In my triangle in Figure 10, there is a dashed line arrow that describes the student's or several students' relation to the teacher's or team of teachers' pedagogical actions, such as giving a lecture. I added this arrow since the student's perceptions of the teacher's pedagogical actions may affect the student's performance in the course. For instance, the studies of Xenos et al. (2002) and Meisalo, Sutinen et al. (2002) showed that the students' perceptions of the lack of assistance from tutors or of tutor teachers' negative attitudes towards giving advice were factors relating to the students' decisions to drop out of the course. The added relation thus emphasises the student's role in the instructional process. The addition of this relation aims at drawing the attention to the fact that there are two active actors present in the process (a teacher and a student). Actions and perceptions of both actors influence the instructional process and its outcome as the studies of Xenos et al. (2002) and Meisalo, Sutinen et al. (2002) exemplify.

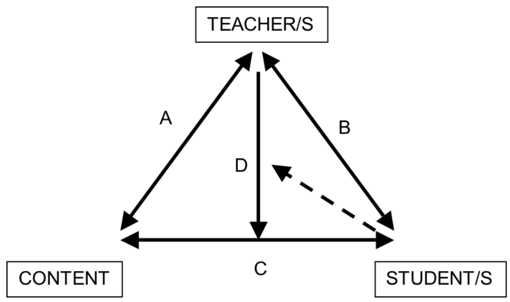


Figure 10 The didactic triangle with an added arrow

The main restriction of Figure 9, however, is that all elements are situated in the context of a single course, whereas an educational system has a much larger scope. To set the triangle in a larger context, each of the three main elements of the didactic triangle can be understood as an instance of some larger entity. Hence, the number of possible viewpoints from which the instructional process can be seen grows significantly. For example, the teacher could be replaced by a team of teachers, the organisation in charge of a degree programme, or even the society at large. Likewise, the student could be replaced by a community of students or even by the citizens of a society. The content node of the triangle can also be seen as a part of a larger entity: the goals for the instruction. Here the goals are understood as a more comprehensive concept than just the taught content. For example, the goals of a course may include the knowledge, skills and attitudes that the student is expected to acquire. Goals can refer to the goals of a single course, a study module, a degree program, or even to the general goals of education that society provides to its citizens. In general, one can view the instructional process on the course level, the organisation level, and the society level. Figure 11 presents these new layers side by side. Note that on the course level the researchers have a choice whether they want to concentrate on an individual student's or teacher's experiences or on the experiences of larger set of students (such as all students in the course) and teachers (such as a team of teachers and assistant teachers in a course). On the organisation level, the community of students refers to a large group of students, such as all computer science majors or even all students at the university.

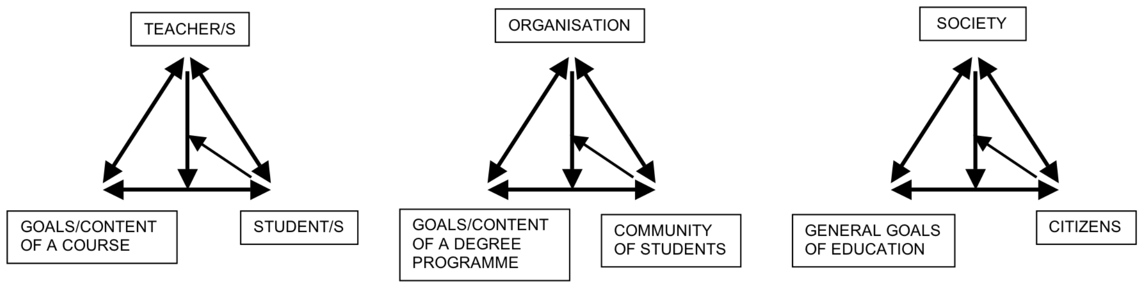


Figure 11 The three levels of the didactic triangle: course level, organisation level and society level

The three viewpoints presented in Figure 11 increase the applicability of the didactic triangle as a base for categorisation of research since it does not confine us to the individual or course level. However, Figure 11 gives the impression that the three levels exist in isolation from each other, even though the different levels actually coexist and interact with each other. If we look at all three levels at the same time, the result could look like Figure 12[6].

The three levels in Figure 11 and Figure 12 can be analysed first level by level and second in terms of how the different levels interact. By applying the level-by-level approach, one can discuss the elements and their interrelations in the same manner as has been done earlier in this chapter when the didactic triangle was first introduced and the individual level (teacher-student-content) was brought into focus. The organisational level (organisation - community of students - goals and content of a degree programme) as well as the society level can be discussed in the same manner. Another approach is to see how different levels combine and intersect with each other. The nodes of the triangles are inclusive so that each student is a part of a larger community of students and hence has a relation to this learning community and also to the larger society. Thereby this three- layered didactic triangle includes also the aspect of community that was introduced in Bergamin's model (2006) as one essential characteristic of the instructional process. Correspondingly, the teacher as a member of the university personnel, has a relation to the organisation and to society, too. The teaching organisation (such as a university) can be seen as a part of an executive system that fulfils the society's educational goals. Therefore, the organisation stands in relation to the society. In the same manner, the content of a course or a larger entity of studies is in relation, for example, to the goals of a degree program and finally to the general goals of the education system.

[6] Note that this is not a geometric shape but a mental model.

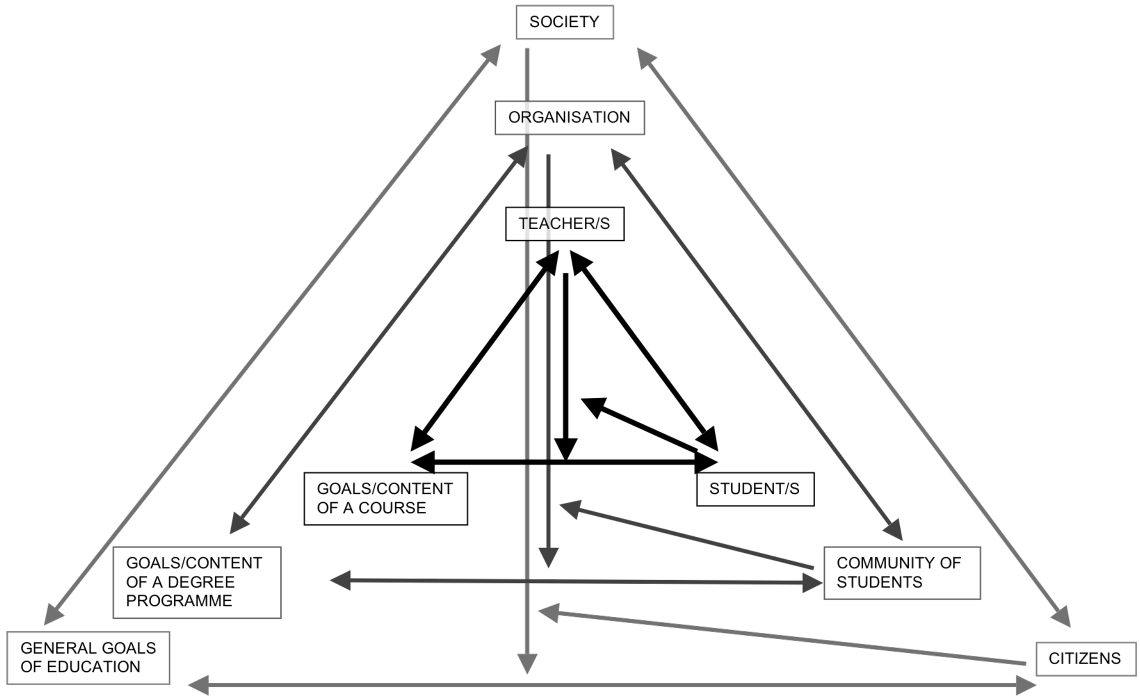


Figure 12 Coexistence of different levels of the didactic triangle

The multi-level didactic triangle has parallels with the "dimension doughnut" approach to the instructional process, which was introduced in the first part of this chapter. First, the dimensions can be found in the three-layered didactic triangle. Dimension A (Student) is represented as the nodes that stand for the student/community of students/citizens in Figure 12. Dimension B (Teaching organisation) is represented as the nodes that stand for teacher/teaching organisation/society. Dimensions C (Content), D (Instructional entity) and E (Level of cognitive process) are represented in the nodes concerning the goals of a course/goals of a degree programme/general goals of education. Dimensions F (Didactic activity), G (Level of concretising) and I (Interaction) represent a large part of with the didactic relation between teacher/teaching organisation/society and students'/community of students' studying and learning process (e.g., the arrow D in Figure 10). Dimension H (Study action) is represented as connections between the student/community of students/citizens and the goals of a course/goals of a degree programme/general goals of education (e.g., the arrow C in Figure 10). Second, a parallel between the triangle and the dimensions is that the three-layered didactic triangle highlights some of the dimensions' categories. For example, categories of the dimensions A (a student, community of students, all students at university) and B (single teacher, team of teachers, teaching organisation) are visible in into the three- layered didactic triangle.

These two approaches to the instructional process, the three-layered didactic triangle and "dimension doughnut", have their own strengths and limitations. The "dimension doughnut" highlights multiple viewpoints and the richness of reality. The dimensions enable detailed examination, for example, of the variety of human interaction during the instructional processes. The limitation of the dimensional approach is that it does not emphasise the interaction between the dimensions and the process nature of the instructional process. The three-layered triangle, in turn, highlights the instructional process in context and the relations between the elements (actors and goal). On the other hand, the dimensions enable closer examination of the elements. For example, different aspects of goals can be divided into several dimensions (as has been done in Figure 7), which enables a more detailed analysis of the goals of instruction.

4.4 Didactic triangle as a base for categorisation of research

In the following paragraphs, the three-layered didactic triangle is taken as a base for development of a categorisation system for educational research. The categorisations that were introduced in chapter two were largely based on the topics of the papers or research approaches used. This section adopts a different approach and concentrates on the *didactic focus* of the study. Didactic focus indicates, which aspects of the instructional process the research aims at studying. I argue that this approach brings out an aspect that is not highlighted in earlier categorisations. Highlighting the didactic focus of research makes it possible to see which aspects of the instructional process have been seen as important or problematic enough to be studied. Moreover, it reveals aspects that may have been overlooked in the past research.

Figure 12 visualises several interactions within the instructional process. However, more and more relations between the levels emerge when the triangles are studied in more. For example, the student has a relation to the organisation as well as to the goals of a single course or a larger entity of studies. If all possible relations between the elements on all three levels are considered, the number of relations grows high. However, not all levels and all relations are equally relevant for the purpose of this chapter, which is to lay down a categorisation foundation that is based on "didactic focus" of the research. For example, the research that focuses on education on the society level often belongs to the field of educational sociology, educational history, or educational politics. Hence, the studies on those fields often fall out of the immediate focus of this thesis.

The following eight main categories are derived from the didactic triangles that are presented in Figure 12. The first three categories are derived from the three main elements of the instructional process, visually expressed as the three nodes of the triangle. The first category addresses the goals and the content of the instructional process. The second and the third category address the actors of the process. The categories four to eight are derived from the relations between the nodes or the node and another relation (the arrows in Figure 12). The resulting eight categories are:

1. *Goals and content*: The focus of the research is on the goals and/or content of the instructional process.
2. *Student(s)/community of students/citizens:* The focus of the research is on one student or the students of one course or the students of some degree program. In the widest context, this category would also discuss citizens of a society. In this context, citizens are discussed as actors and objects of the general education system, which the society provides.
3. *Teacher(s)/organisation/society:* The focus of the research is on one teacher or the teachers of a course, the teachers as a part of an organisation or on the teaching organisation itself. Furthermore, the teaching organisations as a whole arrange the education the society has decided to provide to its citizens.

4. *Student(s)/community of students/citizen – teacher(s)/organisation/society*: The focus of the research is on relationships between actors, for example, the relation between the community of students and the teaching organisation (Arrow B in Figure 10 represents the course level relation between the actors).
5. *Student(s)/community of students/citizens – goals/content*: The focus of the research is on a student's or a community of students' relation to the goals and/or content of the instructional process (Arrow C in Figure 10 represents the course-level relation). This category can further be split into three subcategories:
   5.1 The understanding of and attitude about goals and content that the student(s)/community of students/citizens have
   5.2 The actions (e.g., studying) the student(s)/community of students/citizens do to achieve the goals or learn the content
   5.3 The results of the action (subcategory 5.2 above) of the student(s)/community of students/citizens), for instance, a course's passing rate, a department's graduate rates and the acquired skills and knowledge
6. *Teacher(s)/organisation/society – goals/content*: The focus is on the teachers' or organisation's relation to the goals and/or content of the instructional process (Arrow A in Figure 10 represents the course-level relation).
7. *Teacher(s)/organisation/society – studying:* The focus of the research is on the teacher's or the organisation's/society's relation to the way the students understand the goals/content or the studying and to how the students study (cf. subcategories 5.1, 5.2, 5.3) (Arrow D in Figure 10 represents the course-level relation). This category also includes the pedagogical actions that the teacher/organisation/society does to further students' studying process. This category can be divided into three sub-categories:
   7.1 *The conceptions* of teacher(s)/organisation/society *of students' understanding/attitude on goals/content*.
   7.2 *The conceptions* of teacher(s)/organisation/society of *students' actions towards achieving goals* (e.g., studying)
   7.3 *Pedagogical activities* of teacher(s)/organisation/society
8. *Student(s)/community of students/citizens – teacher's/organisation's/society's pedagogical means to enhance learning*: The focus of the research is on the students'/community of students'/citizens' relation to the teachers', teaching organisations' or society's actions to enhance studying and learning (dashed line arrow in Figure 10 represents the course-level relation).

Each category contains three levels: the individual or course level, the organisation level and the society level. In the following section each category is discussed in more detail by using examples of research that belongs to that category. The examples are drawn from the pool of studies that were presented in chapter two. In addition to providing an example of the research focus that belongs to a particular category, the second reason for using this data is to give an example of how the didactic focus based categorisation system can be used as a tool to analyse, which areas of the instructional process have been less studied. All the studies that were introduced in chapter 2 are now placed into a matrix where the eight categories form one dimension and the level of the discussion (course, organisation/community, society) forms another. The didactic focuses were derived from the research questions or the aim of the research as stated in the publications. If a study had several didactic focuses, it was placed in all corresponding categories. Thus, the same reference may appear in two or more categories.

Since the general research goal of this thesis limited the topics of the studies included in the literature review, there were some categories that were left empty or contained only one or two studies. In these cases, the examples of research that would be placed into those empty categories were picked from general computer science education (CSE) or technology related research and placed in related work cells. Table 5 (page 66) at the end of section 4.5 summarises the example studies and reports that have been placed into each category. Table does not intend to be an extensive and inclusive description of research that has been done within abovementioned fields. Rather, it aims at giving an example of the possibilities the didactic triangle provides for categorisation and analysis of the research based on its didactic focus.

4.5 Categories

Category 1: Goals and content

The goals and content category includes research that analyses, for example, the characteristics of the goals or content of a specific course or a larger entity of studies, such as a degree programme. Another example of research that would fall into this category is research that analyses the relationship between the goals and the content in one level (course, degree, general goals of education) or between different levels. These examples point out that the eight categories that are presented in this section are meta-level categories that include several subcategories. For clarity's sake, the subcategories are discussed in detail, in only two categories. Figure 13 visualises which aspect of the didactic triangle goals and content category focuses on.

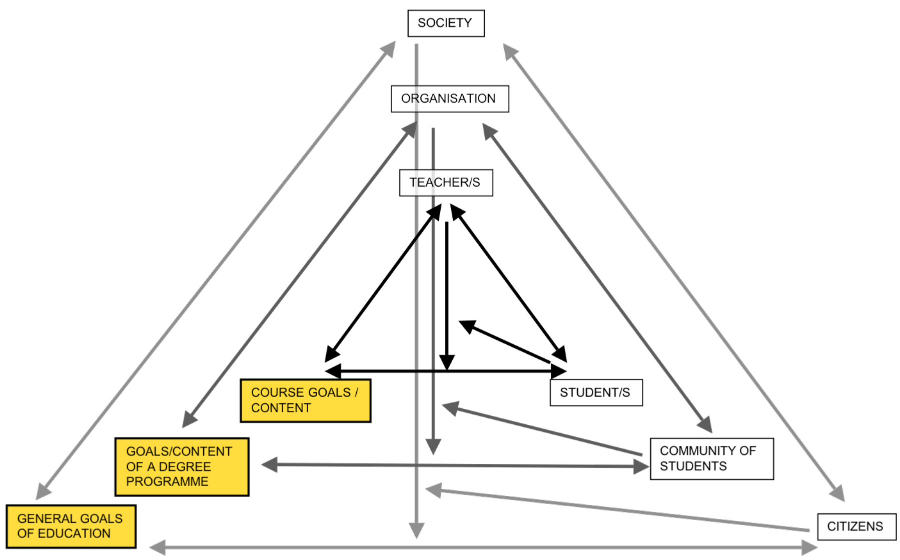


Figure 13 Goals and content as a focus of a category

The literature review (chapter two), which concentrated on studies that focused on difficulties during the instructional process, did not contain any studies that would have solely discussed the goals or the content of the introductory programming courses or a CS degree. The examples of research belonging to this category can be found by expanding the original topic limitation. For example, the study by Stephenson and West (1998) shed light on the programming language choice and key concepts in an introductory programming course (CS1). The study by Reges (2006) also focused on the content of the CS1 course by emphasising the basic skills (e.g., loops, conditionals and arrays) in introductory programming course. These two studies are examples of course level research that focuses on goals and content.

The studies of Tucker, Keleman and Bruce (2001) and Reed, Miller and Braught (2000) are examples of research that focus on organisation level goals and content. The former study highlighted the diminishing role of the theoretical and mathematical ideas in CS curricula. The latter study suggested integrating the development of empirical skills throughout the CS curriculum. The doctoral thesis of Surakka (2005) on the technical skills that graduates from the specialisation in Software Systems need is an example of a study that analysed the goals of a degree programme. The society's viewpoint was strongly present in Surakka's study. Job advertisements were used as one data source in this dissertation and therefore the needs of the industry and the society were highlighted. Thus, in this study the goals were discussed at the society level. Another example of a report that focuses on goals and content from the society's viewpoint is Computing Curricula 2005 (ACM 2005). It is an example of a report that summarises, for example, the performance capabilities expected of the graduates from different computing related degree programs. Thus, this report summarises the general educational goals of different degree programs. Computing Curricula 2001 Computer Science (ACM 2001) on the other hand specifies the goals and content of a degree program on the core topics and course levels.

The computing curricula provided an example of the similarities and differences between research and reports that were found on the organisation and society level. Computing curricula discuss the degree level goals and content. However, an organisation that develops them bases their work on observing the international working life and scientific trends. Thus, the focus of category 1 at the society level is clearly larger than a single organisation. Another example of society level reports are national strategies, for instance, the Finnish Ministry of Education's Regional strategy for education and research up to 2013 (2004) and Development plan for education and research 2007- 2012 (2008), which both state the general goals for technology education, among the other things.

Category 2: Student(s)/community of students/citizens

This category focuses on the student(s)/a community of students/the citizens of a society as actors in an instructional process (Figure 14). This definition is loose and it allows a variety of studies to be placed into this category. First, the category includes research that addresses the characteristics, knowledge, or prior learned skills of students, community of students, or citizens. For example, research that analysed how students' prior academic experiences and programming self-esteem (Bergin and Reilly 2006), cognitive styles (Mancy and Reid 2004) or problem solving ability (Pillay and Jugoo 2005) affects success in a CS1 course belong to this category. As these examples illustrate, the characteristics of the student(s)/community of students/citizens contain a variety of factors some of which are inherent qualities, such as gender, some are academic attainments, such as solid mathematic skills, and some are personal characteristics, such as perceived self-images of the students. Overall, these are the qualities of the students(s)/community of students/citizens that they bring with them to the instructional process.

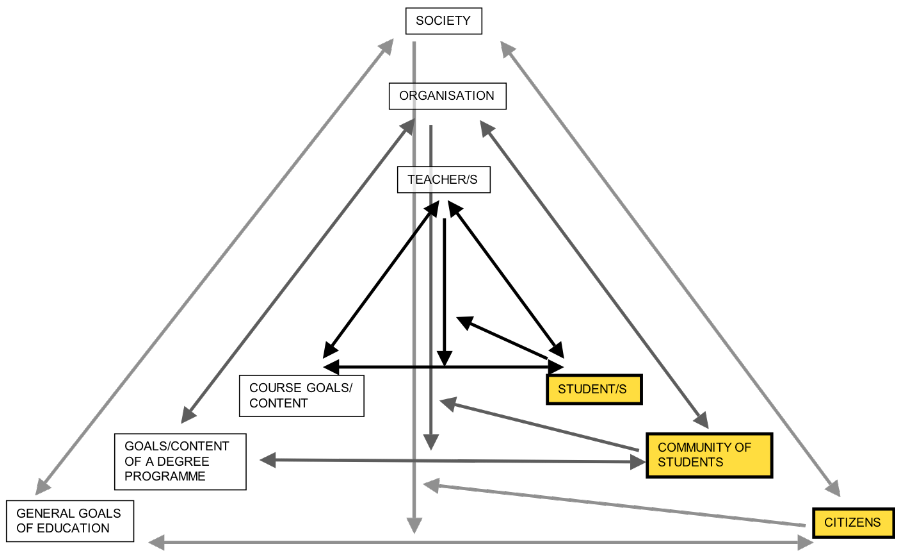


Figure 14 Student(s)/community of students/citizens as a focus of a category

Second, the category includes research that analyses the students' relation to fellow students or the community of students. One focus of the study by Garvin-Doxas and Barker (2004) was the interaction between students. This included, for example, informal discussions before lectures. The data was collected from first year computer science courses but the focus was on community level issues. Therefore, the research is placed on the organisation/community level. Another study with a similar focus was the study by Crenshaw et al. (2008) that focused, among other things, on the students' perceptions of the community of CS students. The study's focus was on general CS studies and it discussed aspects that are community level issues. Therefore, this study was placed in the related work section in the organisation/community cell. This study is an example of research that focuses on the interaction between different level nodes in one vertex of the triangle.

Category 3: Teacher(s)/teaching organisation/society

Category 3 focuses on teacher(s)/teaching organisation/society as actors in the instructional process (Figure 15). These actors are the ones who are responsible for organising and realising the formal education in a society. The level of teacher(s) refers to a single teacher/assistant teacher in a course or a team of teachers who have shared goals, for example, organising a basic studies module. The teaching organisation refers to the abstract institution that provides the resources for teaching and learning. However, teaching organisations often are hierarchical and it is possible to distinguish different levels of the organisation. For example, Helsinki University of Technology (TKK) is organised into four faculties that each consist of departments. Thus, if needed, it would be possible to divide the teaching organisation level further to different levels of teaching organisation. The society level refers here to the structures of the society that are responsible for education. The Finnish Ministry of Education, which is responsible for developing educational, science, cultural, sport and youth policies, is an example of a society level actor.

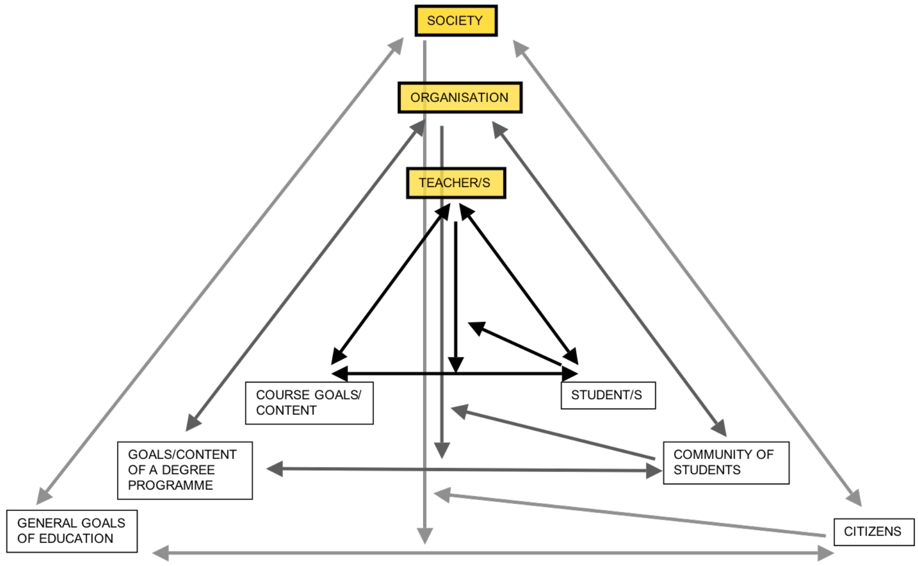


Figure 15 Teacher(s)/teaching organisation/society as a focus of a category

The different focuses within this category include aspects, such as the interactions between actors on one level (e.g., between teachers) or between levels (e.g., the ways a teacher can affect organisation level decisions). For instance, the study by Ben-David Kolikant and Pollack (2004) analysed computer science teachers' relation to their colleagues and highlighted the importance of interaction between the teachers. Another approach is to focus on the characteristics or qualifications of the actors. The study by Gal-Ezer (1995) emphasised the need for qualified educators and proposed a design of the computer science teachers' certification program.

Category 4: Student(s)/community of students/citizen relation to teacher(s)/ teaching organisation/society

This category's focus is on the relationship between the two actors of the instructional process: student(s)/community of students/citizens and teacher(s)/teaching organisation/ society (Figure 16). The relationship does not contain the pedagogical aspect (e.g., teaching, which is discussed later) but the way the actors perceive each other. For instance, the study by Crenshaw et al. (2008) analysed the CS students' conceptions of the members of the department. The study by Foor et al. (2007) also focused on the student's conception of her teachers and the university as a learning environment. The study also illustrated teachers' conceptions of the student and her abilities (as reported by the student). These examples suggest two possible subcategories for this category. The first is the conceptions of students/community of students/society of the qualities and abilities of teachers/teaching organisation/society. The other subcategory emphasises the same relation but in the opposite direction.

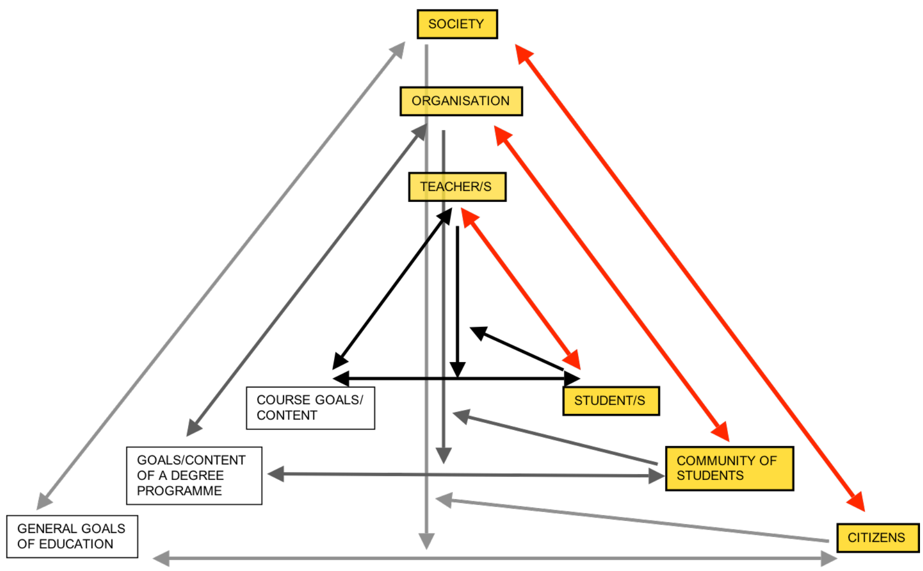


Figure 16 The relation between the actors as a focus of a category

Category 5: Student(s)/community of students/citizens relation to goals/content

This category focuses on the relation between the student(s)/community of students/citizens and the goals and the content of the course/degree program/general goals of education (Figure 17). According to Kansanen (2003) the relation between the students and the content of the course manifests itself as studying. That is, the students' actions when they are striving to achieve the goals, for example, learning some specific skill or completing a degree. However, this relation manifests itself in other ways, too. It can be interpreted as the ways students perceive the goals and content: do they think the goals are worth working for and do they think that they are capable of achieving these goals. The students' conceptions of the content and the different ways of understanding this content are also a manifestation of this particular relation. Additionally, the results of the studying bring yet another dimension to this category. The results of the studying could be, for instance, grades, passing rates, achieved skills and changed attitudes. In order to address the different possibilities the relation between the students/community of student(s)/citizens and the goals and the content of the course/degree program/general goals of education, this category is divided into three subcategories:

5.1 The understanding of and attitude about goals and content that the student(s)/community of students/citizens have

5.2 The actions (e.g., studying) the student(s)/community of students/citizens do to achieve the goals or learn the content

5.3 The results of the action of the student(s)/community of students/citizens (5.2)

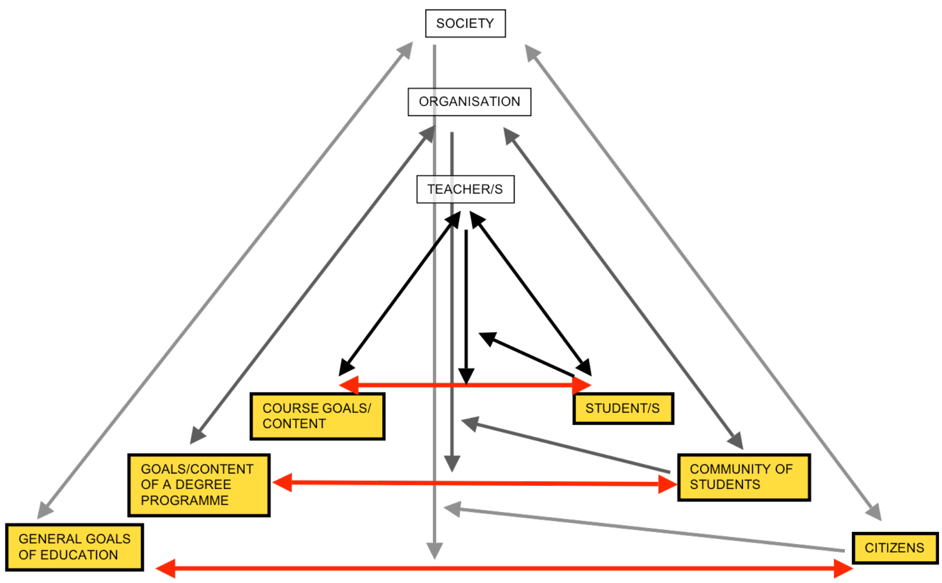


Figure 17 The relation between the student(s)/community of students/citizens and content/goal as a focus of a category

Subcategory 5.1 focuses on research that analyses the understanding on or attitude towards goals and content that students/community of students/citizens have. In a field of computer science education the doctoral thesis by Booth (1992) is one of the first studies that shed light on students' conceptions of nature of programming and programming language. These conceptions are an example of the relation that subcategory 5.1 stands for. This category also includes the research that has analysed the students' problems that derived from incorrect or partial understanding (Spohrer and Soloway 1986) or discussed how students understood some specific programming related concepts (Sorva 2007; Sorva 2008). On the other hand, also the studies that focused on the students' conceptions of the difficulty of the course (Rountree et al. 2004) or on the students' intrinsic goal orientation and task value (Bergin et al. 2005) belong to this subcategory. The studies that focused on the students' comfort level (consisting of factors such as asking and answering questions during the course, anxiety level when working on assignments, perceived competence compared with classmates, and perceived difficulty of assignments (Cantwell Wilson and Shrock 2001) also belong to this category.

Subcategory 5.2 concentrates on research about how students/community of students/citizens act towards achieving goals. Naturally, there are several different types of actions a student can choose to achieve goals and thus the content of this subcategory is diverse. For example, on the one hand there could be research that discusses non-legitimised actions, such as plagiarism. On the other hand, there could be research that discusses the deep and surface approaches to learning. Yet another type of research belonging to this category is that which analyses the concrete content related problems students encountered during studying sessions (e.g., Garner et al. 2005; Robins et al. 2006) or the group dynamic related challenges students face (e.g., Kinnunen and Malmi 2005).

Subcategory 5.3 focuses on the results of the acts that the student(s)/community of students/citizens do to achieve their goals. Just as the students' actions can vary, so can the type of results, too. The results can be, for instance, achieved skills and knowledge measured as received grades or degrees (e.g., Cantwell Wilson 2002), or drop-out rates (e.g., Xenos et al. 2002).

Category 6: Teacher(s)/teaching organisation/society relation to goals/content

Category 6 focuses on the relationship between the teacher(s)/teaching organisation/ society and goals/content of a course or a degree or the general goals of education (Figure 18). This relationship may externalise several ways. On a course level it could mean the way in which the teachers understands the goals and the content of the course or the attitudes the teacher has towards the goals and the content. On a degree level, the relationship could manifest as the degree requirements that the teaching organisation sets for a bachelor's or a master's degree.

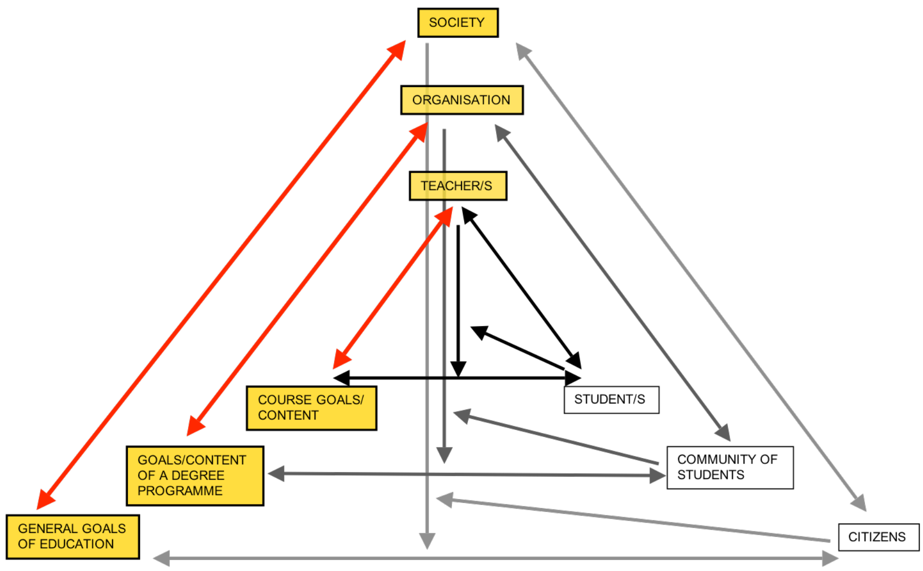


Figure 18 The relation between the teacher(s)/teaching organisation/society and content/goal as a focus of a category

The course level research that belongs to this category focuses on teachers' conceptions of or opinions on concepts and goals and the field of science they teach. For example, the study of how CS teachers' understand 'object-first' belongs to this category (Bennedsen and Schulte 2008). Another example is the research that focuses on the teachers' content and goal related opinions, for instance, which topics are important to teach in an introductory programming course (Goldman et al. 2008; Schulte and Bennedsen 2006).

The study by Gruba, Moffat, Søndergaard and Zobel (2004) discussed the factors that affected curriculum change in computer science departments. The study brought out factors, such as outspoken individuals, academic fashion, financial concerns and student demands were mentioned as influential when it comes to curriculum change. This study's focus is on the teaching organisation's relation to the degree level goals and content. Another example of a similar focus is the study that focused on the process of teachers' assimilation of the new CS curriculum (Haberman et al. 2003). The study by Stein (1999) highlights yet another type of research that belongs to this category. This research proposed a new perspective from which computer science can be seen. Thus, it focused on how the community of teachers (scientists) think about the subject they teach.

The society level research and reports that belong to this category focus on the relation between the society and the goals and content. For instance, the influence of the national strategy can be found in the Finnish framework curriculum for comprehensive school. The study by Lattu (2002) highlighted the idea that curriculum of technology education is a product of a political process. The research then analysed the framework curriculum to see whose voices could be heard in the curriculum. This study illuminates the relation between society level decisions and curriculum level goals.

The second example of society level reports that have a potential to affect the degree and course level goals and content is the feedback reports from the industry and alumni. These sources provide the experience based feedback on what kind of skills and knowledge is currently needed in working life. These previously mentioned examples illustrate the connections between society and a degree, as well as course level goals and content. Many of the reports do not solely rely on research, but rely on ongoing development work and experience based knowledge.

Category 7: Teacher(s)/teaching organisation/society – studying

The focus of this category is on the relation of the teacher(s)/teaching organisation/society to the relation of student(s)/community of students/citizens to the goals and content of a course or a degree or the general goals of education (Figure 19). Next, this complex relation is discussed further. For simplicity's sake in the following paragraphs "teacher(s)" refers to the teacher(s)/teaching organisation/society.

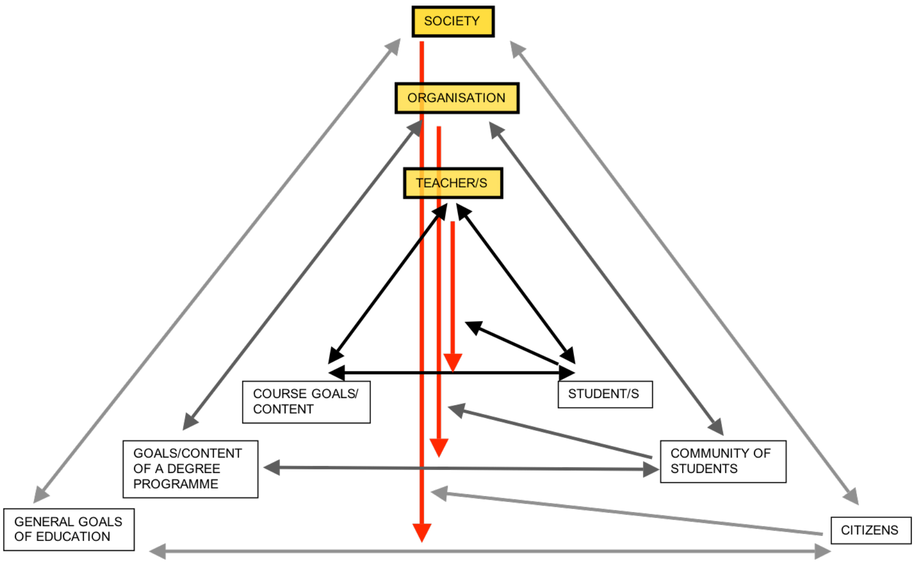


Figure 19 The relation between teacher(s)/teaching organisation/society to the relation of student(s)/community of students/citizens to the goals and content

Previously, category five (the relation of students/community of students/citizens to the goals and content) was divided into three subcategories. These concentrated on the way students understood or felt about the goals and content, the way students acted to achieve the goals and what were the results of their actions. In a similar manner, category seven can be divided into three sub-categories. The first subcategory concentrates on teachers' understanding of the first subcategory (5.1) of category five: what do teachers think about how students understand goals and content or of what students' attitudes are towards goals and content. The second subcategory concentrates on teacher's conceptions of the students' actions to achieve goals. The third subcategory focuses on teacher's pedagogical actions to enhance students' studying process. This is a large subcategory that includes a variety of topics, such as discussions on the pedagogical means used and the learning environments created. The three subcategories are:

7.1 The conceptions of teacher(s)/organisation/society of students' understanding/attitude on goals/content.

7.2 The conceptions of teacher(s)/organisation/society of students' actions towards achieving goals (e.g., studying)

7.3 Pedagogical activities of teacher(s)/organisation/society

Subcategory 7.1 focuses on how teacher(s)/organisation/society understand students' understanding or attitude on goals or content. This category could include studies that about the teachers' conceptions of how useful students find the course, or studies that shed light on teachers' conceptions of how students understand some particular concepts. For example, the study by Haberman et al. (2003) emphasises the importance of teachers' awareness of students' conceptual understanding.

The studies by Lahtinen et al. (2005) and Schulte and Bennedsen (2006) are examples of studies that analysed, among the other things, teachers' conceptions of which course topics are difficult for students to learn. The study by Kinnunen, McCartney, Murphy and Thomas (2007) had a slightly different focus. The research focused on CS teachers' perceptions of student success. This study belongs also to category 7.2 since it highlights teachers' perceptions of aspects that affect students' studying process.

Subcategory 7.3 concentrates on pedagogical actions. The range of studies that belong to this category is wide. On the one hand there is research that analysed which course topics teachers found difficult to teach (Dale 2006). On the other hand, there is research that introduced pedagogical actions to enhance learning. For example, the study by Vagianou (2006) introduced a concept of program working storage to smooth transition of students in introductory programming courses from the end-user stance to the programmer stance. Other pedagogical actions that have been reported were changes in course curriculum to improve students' understanding of course material or general motivation to take the course (de Raadt et al. 2007; Forte and Guzdial 2005).

Category 8: Student(s)/community of students/citizens – the pedagogical means that teacher(s)/teaching organisation/society can use to enhance learning

The last category focuses on the conceptions of the student(s)/community of students/citizens of the pedagogical actions of the teacher(s)/teaching organisations/society (Figure 20). An example of a study that would fall into this category would be a study that analyses the course feedback the students have given to the teacher or the feedback that the graduating students give to the university concerning their studies as a whole. The studies by Cantwell-Wilson (2002) and

Ventura (Ventura 2005) focused on several aspects that can be placed into the didactic triangle. One of the things that they discussed was how students perceived the learning environment and how comfortable they felt in that environment. The study by Sheard et al. (2007) also analysed, among other things, how the students perceived the student-centred, technology-enhanced educational paradigm that the university employed. One focus of the doctoral study by Berglund (2005) was students' experiences of being graded. Therefore, this focus was directed on the students' conceptions of the teachers' pedagogical actions. All these studies are examples of studies that could be placed in the eight category.

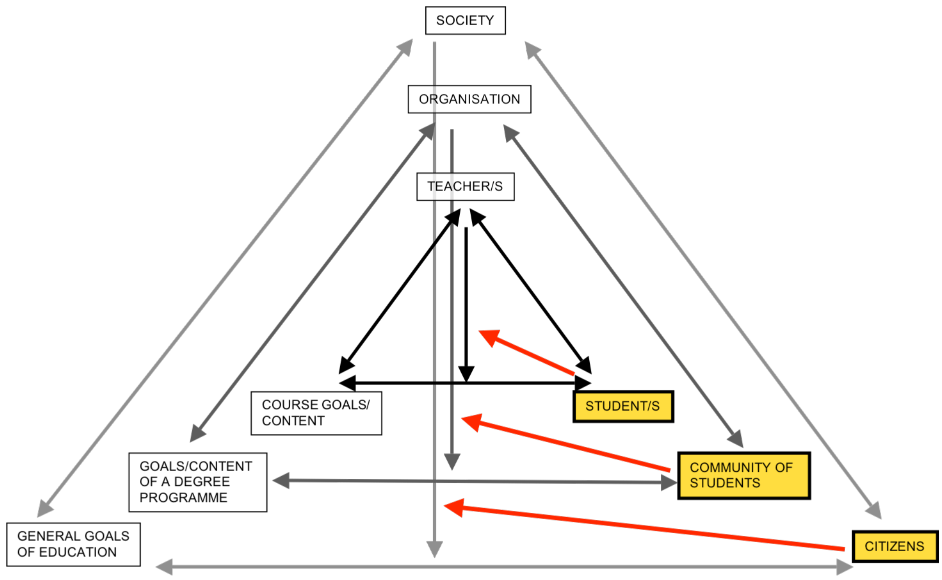


Figure 20 The relation between student(s)/community of students/citizens and the pedagogical actions as a focus of a category

Table 5 Summary of the results of the meta-analysis

	CS1/2 related challenges/difficulties			Related work, CS/ technology education		
	Course	**Organisation**	**Society**	**Course**	**Organisation**	**Society**
1. Goals and content				(Reges 2006) (Stephenson and West 1998)		(ACM 2001; ACM 2005) (Surakka 2005) (Regional strategy for education and research up to 2013 2004) (Development plan for education and research 2007- 2012 2008)
2. Student(s)/community of students/citizens	*(Bennedsen and Caspersen 2005)* *(Bergin and Reilly 2006)* *(Bergin et al. 2005)* *(Byrne and Lyons 2001)* *(Cantwell Wilson 2002)* *(Mancy and Reid 2004)* *(Pillay and Jugoo 2005)* *(Pioro 2006)* *(Ramalingam et al. 2004)* *(Rountree et al. 2004)* *(Simon et al. 2006)* *(Ventura 2005)* *(Wiedenbeck 2005)* *(McKinney and Denton 2004)*	*(Beaubouef and Mason 2005)* *(Garvin-Doxas and Barker 2004)*			*(Barker and Garvin-Doxas 2004)* *(Boyle et al. 2002)* *(Crenshaw et al. 2008)* *(Foor et al. 2007)* *(Goold and Rimmer 2000)* *(Katz et al. 2006)* (Xenos et al. 2002)	
3. Teacher(s)/organisation/society		*(Beaubouef and Mason 2005)*			(Kolikant and Pollack 2004) (Gal-Ezer 1995)	
4. Student(s)/community of students/citizen – Teacher(s)/ organisation/society					*(Crenshaw et al. 2008)* *(Foor et al. 2007)*	
5. Student(s)/community of students/citizens – Goals/content **5.1 Understanding/attitude**	*(Bergin and Reilly 2006)* *(Bergin et al. 2005)* *(Cantwell Wilson 2002)* *(McKinney and Denton 2004)* *(Meisalo, Suhonen et al. 2002)* *(Ramalingam et al. 2004)* *(Rountree et al. 2004)*				*(Crenshaw et al. 2008)* *(Foor et al. 2007)* *(Goold and Rimmer 2000)* *(Katz et al. 2006)*	

	(Simon et al. 2006) (Sorva 2007; Sorva 2008) (Spohrer and Soloway 1986) (Seppälä et al. 2005) *(Ventura 2005)* *(Wiedenbeck 2005)*					
5.2 Studying	*(Bergin and Reilly 2006)* *(Bergin et al. 2005)* *(Cantwell Wilson 2002)* (Garner et al. 2005) (Kinnunen and Malmi 2005) *(Lahtinen et al. 2005)* *(McKinney and Denton 2004)* *(Meisalo, Suhonen et al. 2002)* *(Milne and Rowe 2002)* (Robins et al. 2006) *(Simon et al. 2006)* *(Ventura 2005)* *(Xenos et al. 2002)* *(Xinogalos et al. 2006)*				*(Barker and Garvin-Doxas 2004)* *(Katz et al. 2006)* (Schwartzman 2006) *(Waite et al. 2004)* *(Garvin-Doxas and Barker 2005; Xenos et al. 2002)*	
5.3 The results of the action (studying)	*(Bennedsen and Caspersen 2005)* *(Bergin and Reilly 2006)* *(Bergin et al. 2005)* *(Byrne and Lyons 2001)* *(Cantwell Wilson 2002)* *(Mancy and Reid 2004)* *(McKinney and Denton 2004)* *(Meisalo, Suhonen et al. 2002)* *(Meisalo, Sutinen et al. 2002)* *(Pillay and Jugoo 2005)* *(Pioro 2006)* *(Ramalingam et al. 2004)* *(Rountree et al. 2004)* *(Simon et al. 2006)* *(Ventura 2005)* *(Wiedenbeck 2005)* *(Xenos et al. 2002)*				*(Boyle et al. 2002)* *(Goold and Rimmer 2000)* *(Katz et al. 2006)*	
6. Teacher(s)/organisation/society – Goals/content	*(Goldman et al. 2008)* (Bennedsen and Schulte 2008) *(Schulte and Bennedsen 2006)*	*(Garvin-Doxas and Barker 2004)*			(Gruba et al. 2004) (Stein 1999) (Haberman et al. 2003)	(Lattu 2002)

7. Teacher(s)/organisation/society – studying **7.1 The conceptions of teacher(s)/organisation/society of students' understanding/ attitude on goals/content**					(Haberman et al. 2003)	
7.2 The conceptions of teacher(s) /organisation/society of students' actions towards achieving goals	*(Goldman et al. 2008)* *(Lahtinen et al. 2005)* *(Schulte and Bennedsen 2006)*				(Kinnunen et al. 2007)	
7.3 Pedagogical activities	(Dale 2006) (Forte and Guzdial 2005) (Herrmann et al. 2003) (Kinnunen and Malmi 2005) *(Meisalo, Sutinen et al. 2002)* *(Milne and Rowe 2002)* (de Raadt et al. 2007) *(Schulte and Bennedsen 2006)* (Simon, B. et al. 2008) (Vagianou 2006) *(Xinogalos et al. 2006)*	*(Beaubouef and Mason 2005)* *(Garvin-Doxas and Barker 2005)*			(Andersson and Bendix 2005) *(Sheard and Carbone 2007)* *(Barker and Garvin-Doxas 2004)* (Schwartzman 2007) *(Sheard and Carbone 2007)*	
8. Student(s)/community of students/citizens – teacher's/organisation's/ society's pedagogical means to enhance learning	*(Meisalo, Sutinen et al. 2002)* *(Xenos et al. 2002)* *(Xinogalos et al. 2006)*	*(Garvin-Doxas and Barker 2004)*			*(Barker and Garvin-Doxas 2004)* *(Crenshaw et al. 2008)* *(Foor et al. 2007)* *(Sheard and Carbone 2007)* *(Waite et al. 2004)* (Xenos et al. 2002) (Garvin-Doxas and Barker 2005)	

4.6 Observations and analysis

The didactic focus based categorisation system, which was developed in the previous section, has two benefits. First, it can be used to illustrate what the community of researchers has seen important and relevant enough to study. Therefore, the categorisation system has a potential to reveal something about the state of research in some particular sub-area of educational research. Second, the categorisation system can be used to bring forward less studied research areas and thus it can be used as a tool to find new and relevant research questions. Furthermore, the content of the categories can be analysed in more detail to highlight which aspects, for instance, of student characteristics have been included in the analysis. This information helps to analyse in a more detailed manner, which research questions would benefit from further research.

A quick glance over Table 5 reveals that the vast majority of the research on challenges during the instructional process in introductory programming courses (CS1/2) discussed the challenges from the course level perspective. Research concerning the organisation/community aspects was almost non-existent. The same was true for the society level. This is somewhat surprising, as teachers are generally not completely free to select their course content, the learning resources or teaching methods used. The pressures and guiding principles from the organisation level, as well as available resources may have substantial effects on the students' success, but little research was found concerning this aspect.

Another significant aspect of the material in Table 5 is that there are some rows where there were only few studies placed in cells. The topics relating to course goals and content, students' perceptions of teachers or vice versa, and teachers' conceptions of students' attitudes and students' understanding of goals and content were less studied areas. Additionally, topics teachers' characteristics, qualification, education, conceptions of course/degree goals, content, and students' studying process were analysed only in one or two studies.

It is not possible to make any far-reaching conclusions on the state of research based on the limited pool of studies that were used as a data in this example. However, it is possible to observe some general trends concerning what the community of researchers in the fields of computing education (CE) and computer science education (CSE) have seen as relevant topics to study.

In contrast, there were many studies that focused on the students' characteristics, their studying and the results they gained through studying. Many studies also analysed how students understood the course content. In these studies, the difficulty of the course content was approached from the students' point of view. Studies concentrated on revealing how difficult certain course topics were for students to learn, according to the teacher or the students themselves. However, which course topics were difficult to *teach*, was a much less studied area. Finally, some studies discussed the social or cultural learning environment and the difficulties that related to them. In general, there were not many such studies published in the field of CE or CSE. Furthermore, most of these publications discussed the environment in more general settings than introductory programming courses.

The previous observations are an example of how the categorisation system that is based on a didactic focus can be useful to highlight the areas that are less studied. This knowledge can then be further used to analyse whether those less studied areas would provide new research topics. Another advantage of the didactic focus based categorisation is that, if needed, the studies in different categories can be observed more closely to find out how some research questions were approached. As an example, I include below (Table 6) a list of references of the studies on factors that predicted or explained the students' success in a CS1 course. Out of the 13 studies five focused only on the students' characteristics (category 2) and the results of the studying process (category 5.3). Three studies had an additional focus on the students' conceptions of the course goal and content (category 5.1). This added focus concentrated for the most part on the students' self-efficacy beliefs, concerning their ability to learn programming. Finally, five studies focused also on the actions students undertook to achieve the goals (category 5.2). A closer examination of the actions that were analysed reveals that many studies emphasised general studying preferences such as the deep and surface approaches to learning or aspects of self-regulated learning. The actual actions observed were limited to the number of working hours and game playing during the course. The focus of these studies was on the student and the studied factors were often quantitative by nature. This indicates that there are overlooked aspects of the instructional process that would need closer investigation. Extending the focus, for instance, to students' actual studying related actions (e.g., lecture attendance and usage of learning material/environment) or teachers' pedagogical actions would strengthen the knowledge of the factors that affect students' success and failure.

Table 6 An example of the studies on factors that predicted or explained the students' success in a CS1 course

References	*didactic focus*			
(Bennedsen and Caspersen 2005)	**2**			**5.3**
(Byrne and Lyons 2001)	**2**			**5.3**
(Pillay and Jugoo 2005)	**2**			**5.3**
(Pioro 2006)	**2**			**5.3**
(Mancy and Reid 2004)	**2**			**5.3**
(Ramalingam et al. 2004)	**2**	***5.1*** *(self-efficacy)*		**5.3**
(Rountree et al. 2004)	**2**	***5.1*** *(perceptions on difficulty, work load)*		**5.3**
(Wiedenbeck 2005)	**2**	***5.1*** *(self-efficacy)*		**5.3**
(Bergin and Reilly 2006)	**2**	***5.1*** *(self-efficacy)*	***5.2*** *(game playing)*	**5.3**
(Bergin et al. 2005)	**2**	***5.1*** *(task value)*	***5.2*** *(self-reg. learning)*	**5.3**
(Cantwell Wilson 2002)	**2**	***5.1*** *(comfort level)*	***5.2*** *(work style)*	**5.3**
(Simon et al. 2006)	**2**	***5.1*** *(attitudes to studying)*	***5.2*** *(deep and surface)*	**5.3**
(Ventura 2005)	**2**	***5.1*** *(comfort level)*	***5.2*** *(working hours)*	**5.3**

As a summary, the analysis based on the didactic focus of the research on difficulties during the instructional process in CS1/2 courses highlights that there are some research areas that are clearly overlooked. These areas provide a pool of possible research questions that would help the community of CS teachers and CSE researchers to gain more comprehensive understanding of the instructional process and the challenges during it. For example, CS teachers in general and their conceptions of goals, students and studying processes are areas that would need more research. The importance of finding out more about CS teachers' conceptions and beliefs concerning teaching and learning is evident, as it has been shown that teachers' conceptions are in relation to teachers' approach to teaching (Kember and Kwan 2000). Furthermore, the study by Samuelowicz and Bain (2001) provided evidence that there were fundamental differences in teachers' orientations towards teaching. Teaching-centred beliefs focused on the teaching or teacher or transferring the established discipline knowledge. Learning-centred beliefs focused on learning or the student's role or students' knowledge construction. However, it seems that there are several factors affecting teachers' orientations. The categorised studies have shown that a teacher's orientation to teaching and learning is in relation to taught discipline and the teaching context (Lindblom-Ylänne et al. 2006 June), the higher education environment (Tutty et al. 2008) and to the way teachers understand their subject matter (Martin et al. 2000; Prosser et al. 2005). Therefore, to improve the quality of teaching and further the quality of the learning outcomes, it is important to discover more about CS teachers' conceptions and the factors that affect the teachers' choices in real life situations. The research results of Trigwell et al. (1999) suggest that the teachers' approach to teaching is in relation to the students' approach to learning (deep or surface learning). Thus, in order to encourage students' deep approach to learning one has to apply such approaches to teaching that are in line with a deep approach to learning. However, the first step in affecting the teachers' approaches to teaching is to find out their conceptions of studying, teaching and the goals of the instructional process.

Other research questions, the study of which would contribute to the development of CS education, concern the social and cultural environment. Especially, since several CS departments report high drop-out rates from their degree programs, it would be beneficial to gain more information on the characteristics of the environment and how students perceive it. This would shed light on the reasons behind drop-out rates and thus act as a starting point for development work.

Last but not least, yet importantly, the analysis highlighted that there were very few studies done about the organisation and society levels. However, the university as a teaching organisation affects the teachers in many ways, for example, by giving a certain amount of resources. Therefore, it would be beneficial to do research on under what kind of restrictions and pressures the teachers need to work in reality. These concrete aspects need to be taken into account when planning and implementing a new pedagogical approach or intervention.

5 The feedback loop model – the system theoretical approach to the instructional process

Chapter 4 introduced the "dimension doughnut" and the three-layered didactic triangle that shed light on the multidimensional nature of the instructional process. However, neither of these analysis models is able to address the process nature of instruction. Therefore, this chapter introduces a system-theoretical approach to tackle this aspect of instruction. The first part of the chapter introduces the main tenets of general system theory (GST). In the second part of the chapter, general system theory is applied to the instructional process and the developed feedback loop model figure is introduced. This developed model is explicitly built on the main tenets of general system theory. In the third part of the chapter, the phases of the instructional process and the nature of the feedback during the process are discussed in more detail.

5.1 General system theory

Austrian biologist Ludwig von Bertalanffy (1901-1972) established the field of general system theory (GST) in the 1950s. However, the origins of von Bertalanffy's ideas go further back in time. His work was influenced, for instance, by the works of Norbert Wiener and Alfred Lotka (see e.g. Chen and Stroup 1993; von Bertalanffy 1972). GST is a holistic way of looking at the goal-directed behaviour of complex systems. Its main tenets concern the functions of a system at an abstract level, which makes it possible to apply a system-theoretical approach for making sense of various phenomena across different disciplines. Even though von Bertalanffy developed the original theory in the field of biology, GST has also been applied in other fields such as mathematics, biology and social sciences (von Bertalanffy 1972). For example, Niklas Luhmann used the ideas of GST as a "grand theory" to explain the functions of society and its subsystems, such as education (e.g. Luhmann 2002). On a practical level, it has been suggested that general system theory should be used, for instance, as a unifying general theoretical framework for educational reforms (Chen and Stroup 1993). GST has also been used in a field of organisation studies to highlight the importance of feedback (e.g. Edwards 1992).

The fundamental tenet of system theory dates back to Aristotle: the whole is more than the sum of its parts. This implies that by *focusing on the whole* instead of on individual parts of the system, we can gain a better understanding of the studied phenomenon. Therefore, the focus is not on separate parts of the system but rather on the relationships between the parts. General system theory sees systems as *dynamic* and *cybernetic*[7] by nature: a system transforms itself towards its goals according to the feedback it receives. This emphasises the role of *feedback*. According to the theory, the availability and the use of feedback are prerequisites for *reproduction* of the system (Bai and Lindberg 1999; Chen and Stroup 1993).

Birnbaum (1988) highlighted three elements that all systems have: two or more *interacting components*, *boundaries* and *inputs* and *outputs*. Systems are composed of

[7] The word 'cybernetics' evolved from area of machine control. This first phase of is called first order cybernetics. Later the concept of cybernetics was adopted to social and biological studies. The latter phase is called second order cybernetics (Bai and Lindberg 1999). In this thesis I will discuss the second order cybernetics and use the concepts cybernetics and second order cybernetics as synonyms.

components that interact with each other. A change in one component affects the other components. The components may also be subsystems interacting with each other. Birnbaum used the pool table and an imaginary college as examples. Change of the position of one ball affects the situation of all the other balls on the table. In the school system, the components are more complex and they are not so clearly identifiable objects. Von Bertalanffy (1972) emphasised role interactions by stating that a system is definable only by the interactions of the component elements. Further, interactions or interrelations are conceptual constructs and therefore they cannot be directly perceived. This definition is valid at least in large, complex systems such as universities.

Boundaries define what is a part of the system and what is not. For example, the pool table itself defines what area is a part of the system and what is its environment. When it comes to the school system, the boundaries are less concrete. Inputs are impulses the system receives from the environment that the system processes and returns them as outputs to the environment. In a school system, such as college, the inputs are manifold. For example, students and resources can be seen as inputs that enter the system and educated citizens would then be the output.

Closer examination of systems' interaction with the environment and the relation between inputs and outputs reveals the variety of different types of systems. First, systems can be classified into *open* and *closed systems*. Open systems interact with other systems and the environment whereas closed systems have little or no interaction with the environment. Social systems, such as colleges and universities, are naturally open systems. Birnbaum (1988, pp. 34) describes closed systems as ones that have "boundaries that are relatively rigid and impenetrable and that limit the kinds of interaction that takes place with environment. Input to closed systems tends to be definable, controllable and relatively simple; processing that input can be systematic and scheduled." By contrast, open systems have permeable boundaries and the system interacts with the environment in different ways. Inputs may be diverse and their characteristics cannot always be controlled. Processing the input may be problematic. Open systems are dynamic and nonlinear by nature and the outputs return to the environment where they may become inputs.

Closer examination of the system itself highlights yet another aspect of the systems that helps to make sense of the way systems work and why they produce the kinds of outputs they do. As stated earlier, systems consist of interacting components or subsystems. These components may be tightly or loosely connected. The outputs can be easily controlled and predicted in a system where components are *tightly connected*. In a system where the components are tightly connected, a change in one component always produces the same impact on the other component (see e.g. Birnbaum 1988). For example, setting a wristwatch to a new time always works the same way. If I want to set the time two hours forward, I turn the nub on the side of the watch exactly the same amount each time. In a system where components are *loosely connected*, the outputs cannot be so easily predicted. The impact of a change in one component may vary from time to time since there is no clear connection between the components. When it comes to the social systems, some subsystems may be more tightly or loosely connected than other.

Social systems, such as universities and their subsystems, are loosely connected open systems. A university interacts with the environment and other systems. It receives various inputs from the environment, such as students and resources. The university's interacting components, for example, administration and faculties, are large subsystems themselves. A change in one subsystem may cause a change in another subsystem and, finally, also in the output. However, the relation between the change and the output is not clearly definable.

As stated earlier, a goal-directed system aims at transforming towards its goals. In this process, feedback has a central role. In educational systems, feedback has an established role. Feedback is given and received frequently. Hattie and Timperley (2007) stated that the main purpose of feedback in an educational setting is to reduce discrepancies between current understanding and performance and a goal. In order for the feedback to be effective, it must answer three questions: where am I going (goals), how am I going (what progress is being made toward the goals) and where do I go next (what activities need to be undertaken to make better progress). These three questions work on four levels. First, there is the task level, which concerns how well a task is being performed. Feedback at the task level relates to a criterion that is related to task accomplishment. Second, at the process level, feedback relates to the underlying processes and considers which processes need to be performed in order to accomplish the task or how well the processes are working. Third, at the self-regulation level, the feedback concerns self-monitoring, directing and regulating actions towards the learning goal. Finally, at the self level, the feedback is about the self as a person. The feedback is not tied to a specific task but to personal evaluations of the learner.

5.2 The feedback loop model

In this chapter, general system theory is used as a framework to discuss and develop the feedback loop model in the context of an instructional process. In the following paragraphs, I will elaborate on an abstract description of the instructional process by looking at it from a student's, a teacher's and the organisation's point of view. The objective of the following sections is to discuss general factors that are common to many instructional processes and to address the course-specific issues in the examples.

The instructional process is a goal-oriented activity. Discussing the instructional events from the process point of view puts the focus on the phases of the processes and their interconnections. The instructional process can be described, for example, as consisting of the following phases: setting goals, planning, teaching/learning, processing learning outcomes and feedback (see Figure 21). Several researchers have highlighted the process nature of the instruction (e.g. Lahdes 1997; Meisalo 1985; Meisalo et al. 2003). Here the goals refer to the large set of different kinds of goals that the instructional process can have. For example, the instructional entity from which the situation is observed, dictates the generality of the goals. A degree programme has more general and longer-term goals than a single course. Planning includes choosing appropriate means and seeking for resources to achieve the goals. For example, a teacher may gain information about his or her students' level of knowledge concerning the course's subject matter and take that as a base for his/her planning. There is a vast variety of possible outcomes of the instructional process. However, comparing the outcomes with the goals set earlier, it is possible to understand and highlight the essential outcomes. The feedback is a result of the comparison between the goals and the outcomes. Further, the feedback then acts as a steering mechanism during the following processes. The goals, planning and the teaching/learning phases may be modified based on the feedback. This way of presenting the instructional process highlights the loop nature of the feedback mechanism by showing how feedback affects each phase of the process and hence displays the self-repair nature of the instructional process. (Meisalo 1985; Meisalo et al. 2003)

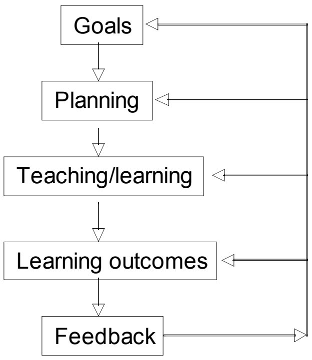


Figure 21 The instructional process (see e.g., Meisalo et al. 2003)

In line with the system theory's tenets, the role of feedback is focal in instructional processes. The cybernetic process is dependent on feedback. What kind of feedback is available and how it is understood and utilised play a central role in how well the instructional process is able to correct itself. Therefore, since the figure of Meisalo et al. (2003) highlights the role of feedback, it is taken as a starting point for elaboration.

The role of feedback is highlighted and enlarged further in the feedback loop model that was developed during this research project (Figure 22). The theory of Meisalo et al. (2003) takes the standpoint that feedback consists of the difference between the goals and outcome. In the following presentation, the foundation of discussion is based on a slightly different standpoint. The role of the feedback that is available after each phase, including setting goals, planning, teaching/studying and processing outcomes is highlighted.

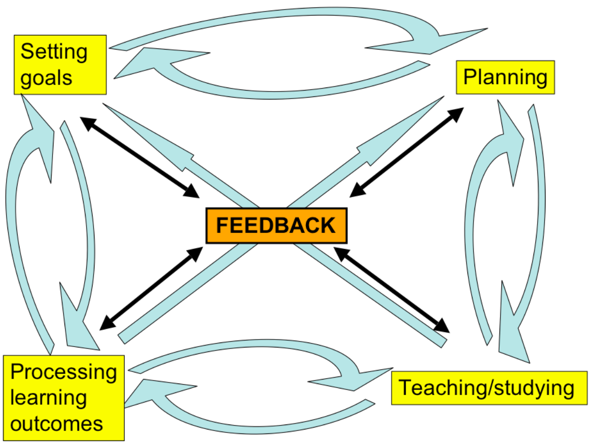


Figure 22 The feedback loop model

Figure 22 can be divided into two figures. The first figure visualises how feedback is available after each phase of the instructional process. The feedback arises from the situation as a result of, for example, self-reflection or discussions with peers. The feedback also actively affects the different phases of the process, hence the two-way arrows (Figure 23).

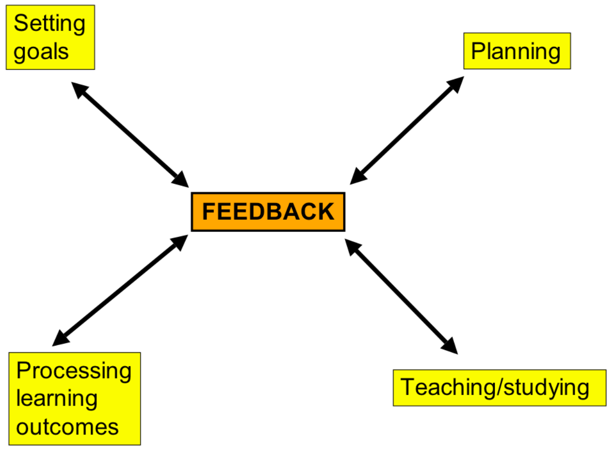


Figure 23 Feedback in the instructional process

The second figure visualises how the instructional process proceeds in time going through all the phases at least once. Figure 24 describes the possible running order of the phases of the instructional process. When discussing one single instructional process, such as a course, from the students' or teachers' point of view these phases follow each other in order. However, correcting loops are likely to happen and situations may emerge where some phases are gone through several times. Correcting loops results from processing feedback that has been received. Feedback is useful only if the receiver analyses it and then decides whether there is a need for further actions. For example, a student in the studying phase notices that he has not understood some central topic of the course. However, he also notices that the teacher assumes that he understands the topic that was taught a few weeks ago. As a result of this notice, the student returns to the planning phase and allocates more time for this particular course to be able to catch up.

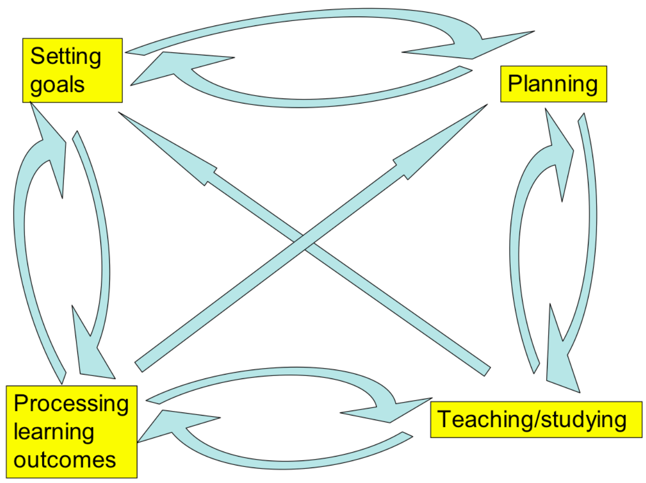


Figure 24 The running order of the phases of the instructional process

Before moving on to dividing the process in to pieces and discussing each phase in detail, it is worthwhile to notice that instructional processes do not occur in a void as independent processes. In contrast, instructional processes form a long-lasting spiral of processes where previous processes affect greatly the following ones (Figure 25). Eventually, the whole curriculum composes of the following processes.

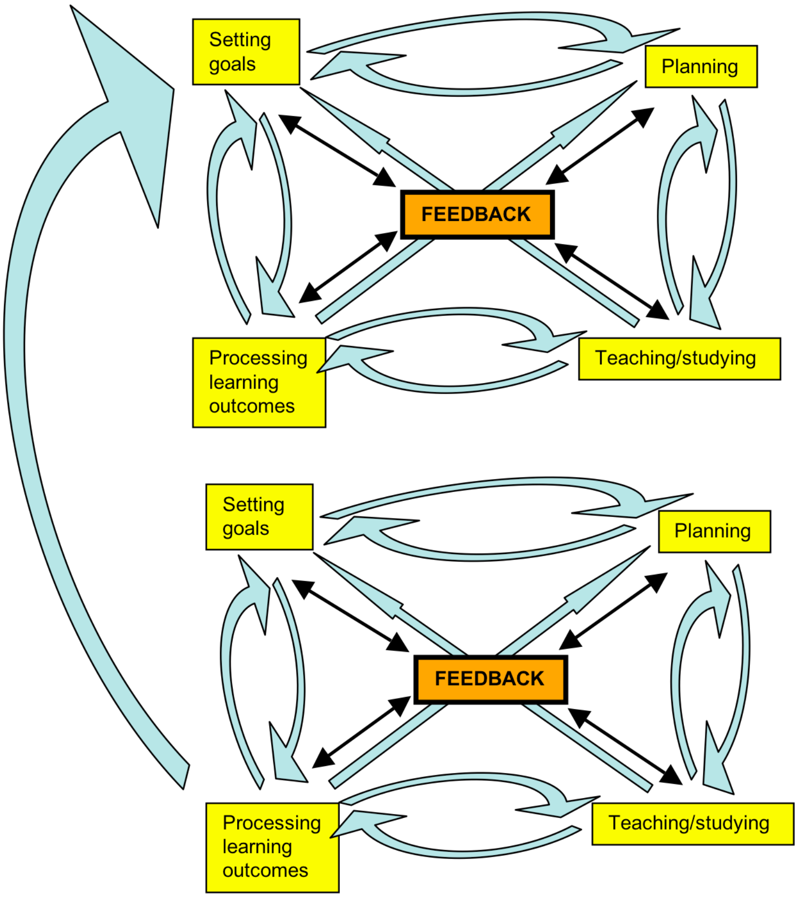


Figure 25 The spiral of instructional processes

In following paragraphs, I will discuss the instructional process from students', teachers' and organisations' point of view. The context of the discussion is university level education. When examples are called for, they are taken from the courses and administration of Helsinki University of Technology. First, I will concentrate on students' instructional process. I will discuss the content of each phase separately. In addition, I will contemplate the possible feedback sources and consequences of feedback after each phase. The same procedure is applied when discussing the teacher's process. When I discuss the instructional process from the organisation's viewpoint, the structure is slightly different. First, I will discuss all the phases of the process in a row and only then contemplate feedback.

The following description includes some aspects that are obvious to anyone who has been in a school. However, in the following paragraphs, these certainties are discussed in a systematic way. The purpose of the description is to give an overview of what kind of aspects and actions each phase could include. However, the main focus of this analysis is not on the content of each phase but the process nature of the instructional process and the role of feedback in the process. Systematic discussion of the phases and the role of feedback during the process make it possible to understand the possibilities the instructional process has to correct its actions towards the goals.

Furthermore, it is important to note that each phase can be discussed on several levels. However, several consecutive and concurrent instructional processes form larger study entities such as degrees. Therefore, each phase in the feedback loop model can be viewed on the course level as well as on the study module or degree level. In Appendix 1, there is a table that summarises all the phases of the instructional process and the meaning of the arrows. In the following sections I will discuss the feedback loop model from students', teachers' and teaching organisations' points of view and at the same time I will construct Figure 22 step by step.

5.2.1 Students' instructional process

Setting goals phase from students' point of view

The instructional process starts with the setting goals phase. The phase may include considering goals on different levels. On the degree level, students set some goals for their studying. These goals may concern the goals for the entire university education; for instance, to get a degree to get the job one has dreamed of. On the other hand, the student may set some goals concerning all the courses he/she is planning to take in one semester. This could include goals concerning the number of study points as well as the skills one wants to achieve during the semester. The student may also set goals concerning one single course. The content of the goals may vary from simply getting a passing grade to getting the necessary credits to more qualitative goals concerning knowledge and skills.

As these examples emphasise, setting goals is a multilevel process, which is closely related to the motives students have. Motives and setting goals can be approached from a general goal preference/orientation point of view (Magee et al. 1998; Niemivirta 2004) or it can be considered from a more specific, course-level goal setting point of view. The former refers to generalised and underlying preferences or tendencies that guide the studying process in general. The latter are more concrete and situational goals students set to achieve a specific outcome. In addition, the course-level goals are strongly connected to the contextualised motives students have. For example, studies from Eckerdal (2006) and Berglund and Eckerdal (2006) demonstrate the variety of motives CS students have.

The goal orientation and general motives students have are significant from the studying and learning process' point of view. Niemivirta's (2004) study by factors that influence students' situational interpretations and the consequences that those interpretations have in terms of task engagement and performance expresses the connection between the types of goal orientations and learning strategies students adopt. By goal orientation Niemivirta refers to individuals' tendencies or preferences for certain types of desired end states. There were five different distinct types of goal orientations found in this study. Mastery-intrinsic and mastery-extrinsic orientations are connected with positive motivational profile, effort and effective learning strategy. Performance-approach-oriented and performance-avoidance-oriented students preferred superficial learning strategies. Avoidance-oriented students, on the other hand, had the most passive profile of learning strategy usage. Therefore, we can conclude that whether the students' orientation is toward gaining a feeling of understanding (mastery-intrinsic), getting a grade (mastery-extrinsic), demonstration of outperforming others (performance-approach), avoiding demonstration of incompetence (performance-avoidance), or trying to get away with as little work as possible (avoidance), it has a logical connection to the general and specific goals students set for themselves. Even though the orientations listed above are perceived to be fairly stable, it does not mean that they do not change over time and different situations.

The role of the context brings the discussion closer to the specific, more concrete goals that students set at the course level. In these more limited contexts, discussing course specific motives that direct setting goals brings another aspect to the focus. As an example, computer science related motives can be found in Eckerdal's (2006) study that reveals several different kinds of contextualised motives novice students have in a computer programming course. Eckerdal found both intrinsic and extrinsic motives. The former included motives, such as programming is fun and challenging, striving for overview and theoretical understanding and to improve problem-solving ability. The latter included personal benefits of programming knowledge. There are certainly similarities and interfaces between Niemivirta's (2004) generalised goal preferences and the motives Eckerdal (2006) found in her study. For example, the mastery-intrinsic orientation could express itself as an intrinsic motive to get an overview and theoretical understanding.

The study by Berglund and Eckerdal (2006) revealed students' motives in an international project-based course that illustrate further the significance of the context in which motives are talked about. They found three motives: academic achievement, project and team working capacity and social competence. Furthermore, each of these three motives was experienced in qualitatively different ways. This study and the study by Eckerdal (2006) highlight course-related, contextualise motives. The topic of the course directs students' short-term motives and therefore affects the goals they set for themselves. For example, let us assume that some student's motive to take a particular course is to gain social competence. This student's goals most probably differ from those of a student whose motive is directed towards academic achievement.

From the student's point of view, one central aspect of studying is the ability to set optimal goals. By "optimal" I refer to the goals that are attainable and reasonable considering students' skills, resources as well as the help available and course requirements. The students' self-efficacy beliefs are closely related to the goal setting and therefore also to the optimal goal setting. By "self-efficacy" I mean the student's conceptions of him/herself as a learner in general and more specifically, as a learner of this particular subject. It is about persons' conception of his/her ability to direct motivation, cognitive skills and relevant functions towards completion of the chosen task (Bandura 1997). How persistently a person decides to work through difficulties and endure unpleasant emotions is affected by perceived self-efficacy. Furthermore, as we know that a long-term bid is a condition of learning something new, the importance of self-efficacy is still highlighted. The conception of self-efficacy is affected mainly by four sources: the student's previous experiences as a learner, comparison with other students, feedback from others and the student's physiological and emotional state (Chowdhury et al. 2002; Wood and Bandura 1989). Since it seems that the process of setting optimal goals is connected to the self-efficacy beliefs, it sets some expectations for the role of the teacher at this phase. For instance, a student who is taking an introductory programming course cannot be expected to be especially good at setting optimal goals for himself/herself. After all, the student does not have previous experience studying this particular subject, nor do most of his/her peers.

Feedback at the goal setting phase

Feedback helps students estimate the attainability of goals. According to the feedback they get, students can adjust their goals to better meet their resources (see Figure 26 arrow 1). There are at least four feedback sources for the student at the goal-setting phase. First, the students can reflect the goals towards their previous experiences. This could include reflection on aspects, such as for how many courses the student has time for in this semester and when the student was studying this subject the last time was it easy or challenging for him/her. Second, peers can act as a valuable source for feedback and information. Peers' experiences give students some information, for example, on whether the course is considered easy or difficult to pass. Third, the teacher may help the students to set attainable goals and estimate the students' own resources. Fourth, the teaching organisation provides feedback in the form of a model timetable and guidelines. By comparing the goals that have been set to the feedback, the students can revise and adjust the goals if needed.

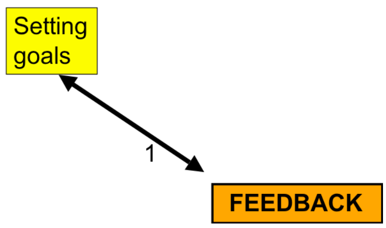


Figure 26 The goal setting phase

Planning phase from students' point of view

The planning phase follows the setting goals phase (Figure 27, arrow 2). This is the phase where students make concrete plans for how they are going to achieve their goals. Plans are concrete strategies for how, when and where studying and learning is going to take place. This includes, for example, how many hours per week the student is allocating to this particular course and what other courses the student is going to be enrolled in at the same time. Similarly to the setting goals phase, the planning phase may concern both long- and short-term plans. Long-term plans may include aspects such as which courses are needed to receive the desired degree and in what order the courses need to be enrolled in (for instance, which courses are prerequisites for other courses). Shorter-term plans may include decisions concerning how many courses the student is enrolling in one semester. At the course level, the plans may be as specific as how many hours the student is planning to allocate for this particular course and how to fit possible laboratory exercises into the student's weekly schedule.

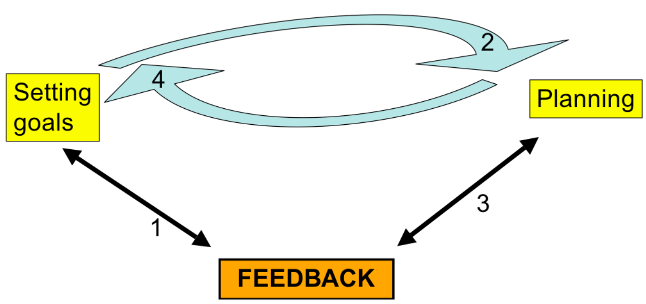


Figure 27 The setting goals and planning phases

Students' study skills and experiences of similar situations are in a central role in the planning phase. The ability to anticipate the time needed for the courses is a skill that evolves as the students gain experience of university-level studies and the various content of the courses. Study skills and the role of experience are highlighted when it comes to first- and second-year students. There are many differences between the study skills required to study at a high school and those needed at a university. Studies have addressed the problems the first-year students confront when they start university-level studies. There might be conflicts concerning students' expectations (of life in university, subjects that are studied and teaching style to name a few) and required study skills including self-regulated learning and managing extensive units of information (Cook and Leckey 1999; Fazey and fazey 1998).

Feedback for student in the planning phase

In the planning phase, students receive feedback from several sources. As in setting goals phase, the students' own reflections on plans compared with previous experiences, self-efficacy beliefs and peers' experiences provide feedback that helps in assessing how realistic the plans are. For example, the knowledge of a particularly time-consuming course spreads from mouth to mouth. The teacher, assistant teachers and course material (e.g., a web page that contains the course schedule) are also sources of information that may help students in the planning phase. The teaching organisation provides feedback in the form of a study guide and a model timetable that offer students guidelines for planning their studies in general. They offer guidelines for when to take which courses and information on the extent of each course. All this is valuable knowledge when students plan the content and the timetable of a semester.

Reflection on plans (Figure 27, arrows 3 and 1) may uncover inconsistencies between plans and goals. For example, the original goals might have been to pass the course with the best grade possible. The plans, however, suggest that there is less time available than would likely be needed to gain the best grade (for example, there is another demanding course that requires a lot of time). As a result of this reflection, the readjustment of goals and/or plans is done. In other words, the student returns to the goal-setting phase (Figure 27, arrow 4), readjusts his or her goals and proceeds again to the planning phase (Figure 27, arrow 2), thus implementing the corrective loop. The other option would be to keep the goals and readjust the plans (for example, make the decision to allocate more time to this course at the expense of other courses).

Studying phase from students' point of view

The concrete teaching and studying phase follows the planning (student moves on to the studying phase Figure 28, arrow 5). This is the phase where the formal and informal teaching and learning takes place. For instance, students take part in the pedagogical activities that the teacher has planned. They may attend lectures, do exercises, read books, interact with the set learning environment (e.g., simulations), or write essays. In addition, there may be a lot of informal studying and learning activities. For example, students discuss course-related issues with each other and learn in this way or use old model answers to figure something out.

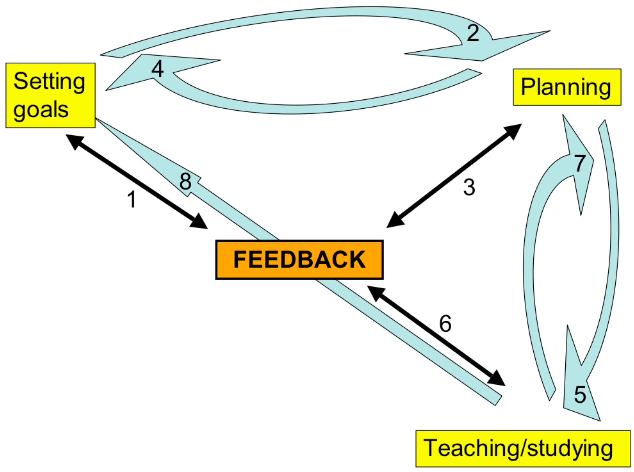


Figure 28 Setting goals, planning and teaching/learning phases

Programming skill is a combination of conceptual understanding and practical skill. Studying and learning computer programming has therefore features that distinguishes it from other studying and learning processes that aim at only learning either conceptual knowledge or practical skill. Eckerdal (2009) concludes in her doctoral study that there exists a mutual and complex dependency between conceptual learning and practice. Learning one enhances the learning of the other or difficulties in learning one may be an obstacle in learning the other. The results of the doctoral study emphasise further the close relation between conceptual and practical learning by stating that practice and conceptual understandings are related through common dimensions of variation.

Traditionally the activities in programming courses support both conceptual learning and practice. For example, the programming exercises are in a central role in all introductory programming courses at Helsinki University of Technology (TKK). In the large-scale course, the conceptual learning is supported by lectures. In the small-scale course, it is supported by small group discussions, assignments utilising concept maps and essays.

Feedback for student at teaching and studying phase

For a student, there is a lot of feedback available during the studying phase (Figure 28, arrow 6). First, there is the feedback from self-reflection. As the students are studying, they receive constant feedback on how they are doing: “Do I understand what the lecturer is saying?”, “Do I know how to do these exercises?” and so forth. Peers are another source for feedback. Seeing how one's peers are doing and one's own achievements to theirs gives feedback on how one is doing compared to others. The teacher is also a source of feedback. The feedback from the teacher may be direct and explicit. For example, when students are doing exercises the teacher may explicitly say how the student is doing. Exercise grades/points are a form of a feedback, too. Full points from an exercise tell the students that they are doing fine.

Feedback from this phase has many implications for students’ studying process. First, it may act as an impetus for revisiting plans (Figure 28, arrow 7). For example, feedback may imply that the student needs to spend more time with the learning material in order to attain goals set earlier. On the other hand, feedback and experiences from the teaching/learning phase may imply that the goals need to be readjusted (Figure 28, arrow 8). For instance, an enthusiastic teacher might have been able to arise interest in the subject matter and now the student has more motivation to study and wants to set his or her goals higher. This, in turn, may have a positive effect on planning and learning phase.

Processing learning outcomes phase from the students' point of view

In the processing learning outcomes phase, the results of the process start to emerge (Figure 29). There are several kinds of learning outcomes. On one hand, the learning outcomes can refer to knowledge and skills and on the other hand to attitudes and interest in the topic. The learning outcomes already start to emerge during the instructional process as the studying process proceeds. On the course level, the emerging learning outcome could be, for instance, the ability to write a while loop.

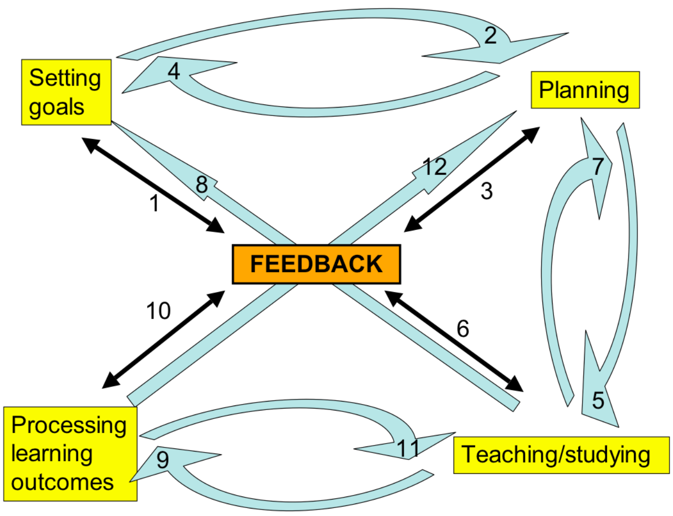


Figure 29 Setting goals, planning and teaching/learning and processing learning outcomes phases

On the other hand, there are learning outcomes that are visible only after the whole instructional process has ended. At the end of the course, it is possible to summarise what has been learned. For instance, at the end of the course, the learning outcome could be an ability to design and write a small program. By comparing learning outcomes and goals, the student gets an idea how well he/she did compared to goals he/she had set earlier. This comparison is the one that helps to analyse the learning outcomes. Comparison between learning outcomes and goals forms a basis for examining other parts of the learning process as well. Once the learning outcomes unfold, the student can start reflecting on what issues led to this end-result. For example, the student could think about what let him/her to succeed so well on some course. Was it, perhaps, the interest in the subject or the time he/she spent on studying? This reflection can then serve as a starting point for the next learning process. The same mechanism works when summarising learning outcomes after a larger study entity.

Another type of learning outcome would be skills or attitudes that will be apparent only a long time after the course is finished. For example, there might be some skills that are taught through many courses and students will gradually become more skilled. This types of skills are not apparent after the first course or two since they evolve gradually.

Feedback for students at the processing learning outcomes phase

In the processing learning outcomes phase, the feedback sources are similar to those in other phases. Students may reflect on the outcomes and compare them to the goals they had set for themselves at the beginning of the process. Large deviations from the goals give students the message that the course went better or worse than they expected. Students may also compare their own outcomes to other students' outcomes and conclude from that how they are doing compared to others.

The course teacher and assistant teachers may also help the student to analyse the learning outcomes by giving feedback. The exercise points and grades from the assignments already give students feedback during the course. For instance, on the course level, as students successfully make an assignment where while loop is needed, they get feedback (exercise points) that they have achieved some sub-goals of the course goals (Figure 30, arrow 10). The teaching organisation gives feedback, too. However, this feedback concerns larger study entities. For example, students may compare how their studies proceed compared with the model timetable or the guidelines concerning the ideal number of years where students should receive their degree. Based on the feedback and situation, students may either revise their current instructional process or use the feedback during the next instructional process. If some learning outcomes already become visible during the process and the feedback is available, students may make changes to study activities (Figure 30, arrow 11), their plans (arrow 12) or even adjust the goals (arrow 13).

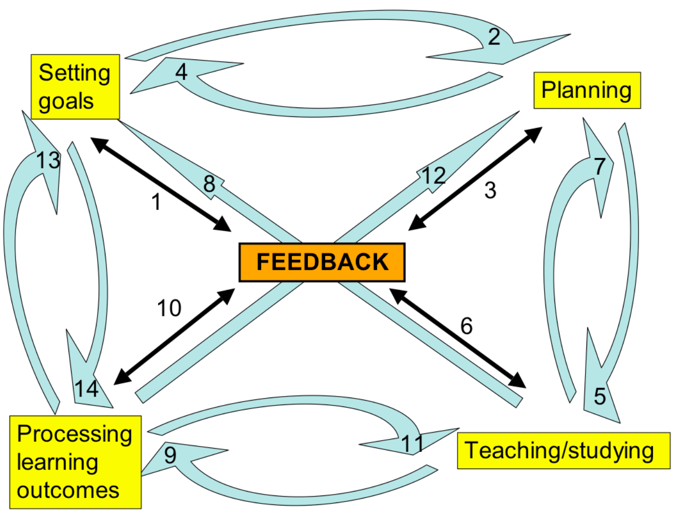


Figure 30 The whole feedback loop model

At the beginning of the new instructional process, students may look back at the previous instructional processes they have gone through (Figure 30, arrow 14, note that this last step completes the reconstruction of Figure 22. The outcomes of those processes give students some idea what kind of learners they are, how good they are at some particular subject and what kind of goals would be attainable for them when they enrol in the next course.

5.2.2 Teachers' instructional process

Setting goals phase from the teacher's point of view

The teacher's point of view of the goals goes back to the assessment of the needs. When a new course or study module is created, one of the first things to do is the analysis of which skills and knowledge students need in their future studies and profession. Analysis of the students' knowledge level when they enter the course or the degree program is another important piece of knowledge for the teacher. This groundwork is an important part of setting goals phase (see e.g. Dick et al. 2004).

As a member of a community of scientists and teaching organisation, the teacher has at least two sources he/she can derive the goals from. On one hand, there are the goals relating to the field of science: to introduce the field of science to newcomers and help them to become a member of the community of scientists. On the other hand, there are goals related to the students' development as a professional, who need many working life skills, both content-related and not. Wenger's (1998, p. 263) definition of education emphasises the importance of building of an identity and becoming a part of community of practice. *"Education, in its deepest sense and at whatever age it takes place, concerns the opening of identities – exploring new ways of being that lie beyond our current state ... education should be addressed first and foremost in terms of identities and modes of belonging ... and only secondarily in terms of skills and information."* This definition of education emphasises the need to highlight identity building and becoming a part of the community of practice in the setting goals phase. These aspects bring out the different types of goals that the teacher must consider: short and long term goals, subject-specific and generic working life skills-related goals.

The teacher's role in the goal-setting phase is to consider what the general goals of the educational system could mean in the context of the course she/he is going to give. The teaching organisation provides the general goals for the instructional processes in the form of mission statements and a list of general goals. It is then the teacher's task to interpret them and divide the general goals into smaller, more concrete and attainable short-term goals that are relevant to the course. From this point of view, the teacher is, in a way, a mediator between organisation-level general goals and course-level goals. Once the course-level goals are set, the teacher can start thinking about how they could be further divided into smaller and more easily attainable forms. That is, what are the sub-goals of exercises and assignments. Therefore, in the end, there is a hierarchy of goals: general level goals, course level goals and exercise level goals. The higher-level goals are often general and abstract whereas the lower-level sub-goals are more concrete.

There are other issues besides the curriculum and content expertise that guide the teacher's goal setting phase. Conventions and earlier experiences guide the teacher in his/her deciding which goals are high in the priority list. The general goals of the curriculum give the teacher freedom to emphasise some aspects of the subject at the expense of others. Therefore, the goal-setting phase in the teaching process depends not only on the official curriculum and the academic community the teacher is part of, but on the teacher's experience and personal preferences, too.

The relation between the teachers' and students' setting goals phases is important. One role of the teacher is to help students to set attainable and reasonable goals. This can, of course, be done in several ways depending on the size of the course and skills and knowledge of the students taking the course. Even though studying at the higher-education level is self-directed, a teacher can still give some general guidelines for the ones who need it. Especially when we are talking about introductory courses, such as CS1, the teacher's role in goal setting phase is important. Most students do not have any previous knowledge of computer programming. Hence, students do not have an accurate picture of what kind of goals are attainable for them considering their background and resources. Therefore, the students need a teacher's help in setting their goals.

Feedback for the teacher at the setting goals phase

For teachers, it is possible to already receive the feedback at this early phase of the teaching process. The teacher him-/herself is an important source of feedback. A teacher's previous experiences with a course and the same or a similar student group provide feedback that the teacher can use to reflect the current goals. For instance, the teacher can reflect the goals against the level of the educational system (e.g., can you expect a high school student to attain these goals?), the learning outcomes of the previous year, the skill and knowledge level of the students who enrol in the course this year and the teacher's experiences and conceptions of whether the goals are attainable. Student feedback from the previous years' courses is also a source that the teacher may use when he/she is reviewing the goals.

Another source of feedback is other teachers. Sharing experiences with other teachers and taking part of the course development work provides a forum for giving and receiving feedback. The teaching organisation may collect student feedback concerning single courses and larger study entities. The organisation can then distribute the results to the teachers who can use it as a feedback source. Depending on the feedback and the teachers' scope for actions, teachers may revise the goals.

Planning phase from teachers' point of view

In the planning phase, the teacher plans how the general degree, study module-level and course-specific goals set earlier can be achieved. This planning may include study module-level planning with other teachers and administration, such as plans concerning in which order the two or three first introductory and basic-level courses should be given, what is the exact content of each of these courses, or what kind of teaching and assessment methods could be used in those courses so that they form a whole.

At the course-level, the teacher may do the planning alone or together with the teaching assistants. The target of the planning phase is to decide on the means to achieve the course goals as well as smaller sub-goals of the course. This phase includes a set of different types of tasks. At one end may be planning and preparing the content of the lectures, the exercises and the type of assessment that is used in the course. At the other end, there may be concrete tasks such as reserving the lecture hall. The content of this phase naturally varies from course to course. The scale, as well as the content, of the course affects the planning phase. For example, in a large-scale course, the timetable and the exercises need to be planned and prepared early whereas in a smaller-scale course it might be possible to do some of the planning during the course.

The teacher has some limitations that s/he has to take into account when planning. The resources limit the options for pedagogical activities. For instance, the budget of the course may restrict the number of lectures and exercise sessions in a course. On the other hand, in another course the teacher may have more freedom and decide to use teaching and studying methods, such as problem-based learning, that are more expensive to organise.

The teacher's experience and pedagogical competence play a central role in the planning phase. The essence of the planning, for the teacher, is to think how to best scaffold the students' studying process. Teacher may think about aspects such as how the course-level goals can be concretised, what to emphasise at the expense of other subject entities, what would be a good way to present new concepts, what kind of activities students will be doing and when and what kind of feedback to give to students and how. The teacher's knowledge of the students and their background knowledge are essential in the planning phase. Based on that, the teacher can start to plan how to best meet the goals of the process. In summary, the plans concretise previously set goals and state explicitly how teachers have planned to achieve them and how the attainment of the goals will be assessed.

Feedback for the teacher at the planning phase

The feedback for the teacher at the planning phase comes from several sources. First, teachers may reflect on their experiences from similar courses. To what degree did the plans prove to be viable last time? Another source of feedback is other teachers. Discussions with colleagues can provide valuable feedback concerning the plans. For example, how did other teachers organise a similar course? What kind of difficulties did they encounter and how did they handle them? The student feedback that the teaching organisation or the teacher him-/herself collects may also provide feedback that can be utilised at the planning phase.

The feedback the teacher receives might suggest that goals need to be revised. For example, the goals might prove to be too ambitious with regard to the students' output-level or the resources available. On the other hand, the goals may be appropriate but the plans are not adequate. The feedback from the planning phase may result in rethinking and adjusting the goals and/or the plans. Thus, the feedback evokes the correcting measures and sets off the self-repairing process of the instructional process.

Teaching phase from teachers' point of view

After the planning phase, the teaching phase follows. The variety of different kinds of teaching situation is great, ranging from face-to-face teaching in small groups to providing an online learning environment. In any case, this is the phase where the plans are put into action, were it in the classroom or providing the learning material and assignments to the students via the Internet. Scaffolding the studying and learning process is the core of the teaching profession.

Studies of higher education indicate that teachers have different kinds of orientations toward teaching and learning that reflect teachers' beliefs about desired learning outcomes and the nature of knowledge. Orientations have a concrete influence, for instance, on the teacher-student interaction during the instructional process and how the students' existing conceptions are taken into account. Studies suggest that teachers possess either teaching-centred or learning-centred orientations toward teaching and learning. The study by Samuelowicz and Bain (2001) provided evidence that there were fundamental differences in orientations. Teaching-centred beliefs focused on teaching or the teacher or transferring the established knowledge of the discipline. Learning-centred beliefs focused on learning or the student's role or the student's knowledge construction. The study by Trigwell, Prosser and Waterhouse (1999) showed that there was a relation between teachers' approach to teaching and students' approach to learning. Students, whose teacher's approach to teaching was teaching-centred, were more likely to adopt a surface approach to studying. Conversely, students, whose teacher's approach was learning-centred, were more likely to adopt a deep approach to learning. Further, students' deep and surface approaches to learning are interrelated with the quality of learning outcomes (Trigwell and Prosser 1991). There is also a study on university teachers' conceptions on what learning is and how learning is achieved (Bruce and Gerber 1995). There are some appealing parallels between the teachers' conceptions on learning and how learning is achieved and teaching orientation. For example, the study by Prosser and Trigwell (1999) suggest that teachers' conceptions of teaching and learning are often related and co-dependent. Figure 31 summarises some of the main results of the previously mentioned studies and shows the supposed (dashed-line arrows) connections between teachers' conceptions, orientations and approaches.

(Bruce and Gerber 1995) According to teachers' conceptions **learning is about**	(Bruce and Gerber 1995) According to teachers' conceptions **learning is achieved by**	(Prosser and Trigwell. 1999)	(Samuelowicz and Bain 2001) **Teaching orientation**	(Trigwell et al. 1999)	(see, e.g., Marton and Booth 1997) **Student's approach to learning**
Acquiring knowledge	Application of study techniques, achieved by individuals	→	Teacher-centric (imparting information)	→	Surface approach to learning
Absorbing and understanding	Application of study techniques, being actively engaged in learning	→	Teacher-centric (transmitting structured knowledge, providing and facilitating understanding)	→	
Development of thinking skills, ability to reason	Individual effort or participation to social interaction (not necessarily 'observable' actions)	→	Learning-centred (helping students develop expertise)	→	Deep approach to learning
Developing the competences	Experiential process, experience with 'real life' problems	→	Learning-centred (helping students develop expertise)	→	
Changing attitudes, beliefs, behaviour	Experiencing alternative ideas etc. through discussion or experiencing contexts that are novel to the learner	→	Learning-centred (negotiating understanding, encouraging knowledge creation)	→	
Engaging in participative pedagogical experiences	Students in different situations learn differently	→	Learning-centred	→	

Figure 31 Connections between teachers' conceptions, their teaching orientation and students' approach to learning

The previous studies suggested that teachers' conceptions may have an important part in actual teaching phase. However, there are several other factors besides teachers' conceptions that affect their orientations. Studies have shown that the orientation toward teaching and learning is related to the discipline and the teaching context (Lindblom-Ylänne et al. 2006 June) and to the way teachers understand their subject matter (Martin et al. 2000; Prosser et al. 2005).

Feedback for the teacher at the teaching phase

For the teacher, there are several forms of feedback and channel to get it from at this point. The teacher may reflect on the successfulness of the teaching. Teachers' own gut feelings may be confirmed or disproved by the feedback they get from students. For instance, during lectures, students' questions and actions give the teacher feedback concerning how well the students are following the topic of the lecture, how motivated they are and how well the teaching methods that are being applied are working.

The feedback that is received can be used to evaluate both the plans and the goals of the course. First, the feedback from the teaching situation gives some information on how well plans are working. Based on that feedback, the teacher may go back to the previous phase and readjust the plans if needed. Feedback from the teaching phase also hints whether the goals of the course will be met. It is possible to use feedback from the teaching phase when returning to refining goals. For example, the teacher may notice that the course is far too difficult for the students and decide to lower the difficulty level of some of the goals.

Processing learning outcomes from the teachers' point of view

The last phase in the teaching process is processing learning outcomes. Learning outcomes can refer to many different types of outcome: short-term outcomes, such as what students have learned after few lectures/exercises. In addition, there are also long-term learning outcomes, such as what students have learned after a whole course or a study module. Learning outcomes can refer to knowledge, skills and attitudes. The course's passing rate and distribution of the grades are also one form of outcome.

Feedback for the teacher at the processing learning outcomes phase

In the processing learning outcomes phase, the teacher's own experiences of how the course as a whole went is one source of feedback. The teacher may compare the learning outcomes with the goals set earlier and analyse to what degree the goals were achieved. The teacher may also compare the statistics on grades and passing rates of the course to the statistics of the previous years' courses or other teachers' courses and conclude, based on that, how successful the course was. There may be several factors that cause deviations and it is the teacher's task to analyse the reasons and whether they call for action. Large differences may, for instance, be a sign to the teacher that the difficulty of the work or the workload deviates from the other courses and the extent of the course or its assessment criteria need to be reconsidered.

In a middle of a course, the emerging learning outcomes may serve as a good source of feedback for the teacher. For example, in a course where students do several exercise rounds during the course, the teacher receives feedback by surveying exercise points at different times during the course. Based on the exercise points, the teacher may conclude whether students understand the course material and how their studying process is going.

Student feedback on the course is usually collected at the end of the course. Therefore, the students are an important source of feedback that can help the teacher to analyse not only the learning outcomes, but also the whole teaching and studying process. The students' feedback from several years provides the teacher data from which he/she can see whether the changes he/she has made over the years have made a difference, for instance, to the students' attitudes or their satisfaction with the course arrangements.

The teaching organisation may provide feedback in the form of rewards and recognitions that are based on good learning outcomes in a course or persistent development work the teacher has done over the years. For example, a Teacher of the Year reward can be seen as an organisation's way of giving feedback on good quality teaching. Another source of feedback is the organisation's official goals concerning the number of study credits and degrees. If the course has a high drop-out rate, the learning outcomes are not in line with the teaching organisation's goals.

Based on the feedback in the processing learning outcomes phase, the teacher may adjust the current or next teaching process. For example, if the emerging learning outcomes indicate that students are having difficulties with some of the key concepts of the course, the teacher may decide that he/she needs to spend some more time elaborating on these central concepts in his/her lectures. This decision leads to adjustments to the plans and teaching phases. At least in theory, very poor emerging learning outcomes during the course may force the teacher to lower the level of the goals, too. The teacher may decide, for instance, to focus on some skills and concepts and pay less attention to some other topics in order to attain at least the minimum goals.

5.2.3 Teaching organisation's instructional process

The teaching organisation is a large, often hierarchical, system. There are several different levels of organisation from which the instructional process can be discussed. For example, the Ministry of Education has its own goals and functions, which affect the functions of Helsinki University of Technology (TKK). The organisation of TKK consists of a council, faculties and departments (Figure 32).

The universities of Finland have goals concerning teaching and research. In the following discussion, I will focus on teaching. The relevant aspects in this context are how the teaching organisation takes care of the goals and content of the instruction, measures to support teaching and learning and resources. In the following paragraphs, I will concentrate on looking at the goals and content in the light of the process model. Unlike the process description at the student and teacher-level, at the organisation-level, I will first discuss each phase and only after that will I focus on the feedback that the organisation receives.

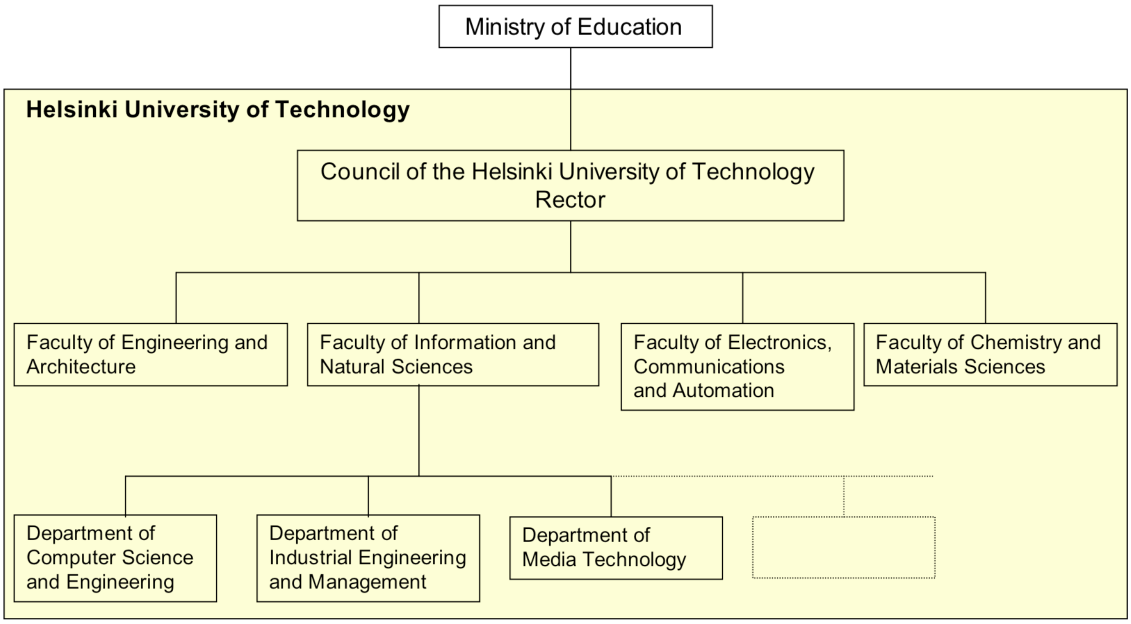


Figure 32 Organisation chart of the Helsinki University of Technology

Setting goals phase from the organisation's point of view

Each level of the organisation has its own tasks and goals. The more high-level the organisation, the more general goals it has. As a part of the Finnish Government, *the Ministry of Education* is responsible for developing education, science, culture, sport and youth policies and international cooperation in these fields. Its task is to develop conditions for education, expertise, lifelong learning, creativity and citizens' social participation and well-being. To meet these goals, the Ministry of Education sets general goals for higher education. TKK has its own more specific goals that are in line with the goals set by the Ministry of Education and the Universities Act.

The Council of Helsinki University of Technology (and the Rector as a central part of it) is the highest decision-making body in the University. The Council's role is to provide prerequisites for University to function according to its mission. The Council decides on the objectives for the degree programs. However, the Council does not commit on the content of the studies as such, even though it is responsible for the uniformity of the educational activities in the university. The Council sets general quantitative and qualitative goals for the university. The quantitative goals concern the number of the degrees the university is going to award the following year. The qualitative goals can be found in the mission and the strategy statements. For example, the Council states in the Degree Regulations the objectives for Bachelor's and Master's degrees. The Council also approves decisions that concern the University's finances and other far-reaching plans. Therefore, it also makes decisions concerning the guidelines for allocation of appropriations.

The f*aculties* set both quantitative and qualitative goals for their actions. In addition to ensuring enough students graduate, each faculty is also responsible for the content of Bachelor's and Master's degrees that are under their supervision. The faculty is responsible for producing professionals within a certain discipline, for example, computer science and information technology. Therefore, the faculty manages the departments and sees that the education they provide results in professionals of that discipline. The Degree Regulations of the Helsinki University of Technology (TKK) set goals for the Bachelor's and Master's degrees and therefore act as a guideline for faculties when they plan the content and structure of their degrees. In Figure 33 is an example of the structure of Bachelor's and Master's degrees in Helsinki University of Technology (TKK). Faculties decide which courses different modules consist of and therefore define a basis for the more concrete goals of the degree.

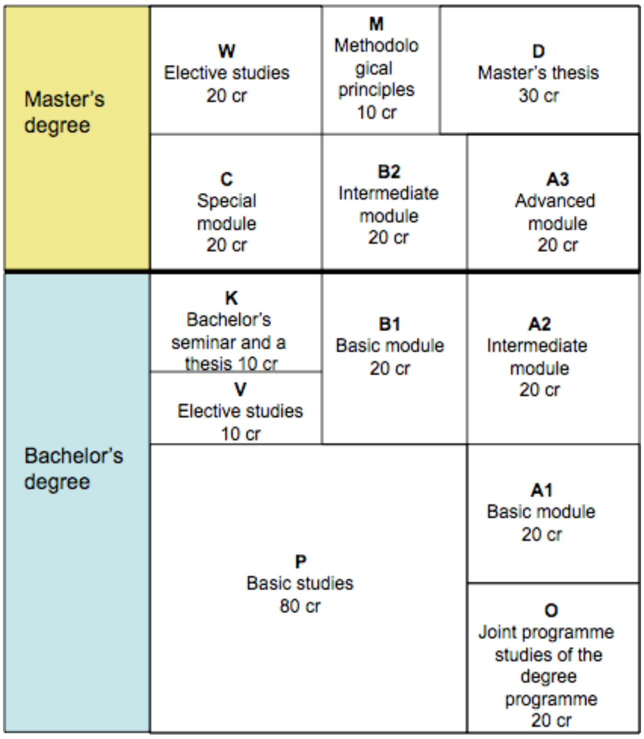


Figure 33 Structure of the Bachelor's and Master's degrees at Helsinki University of Technology

A *department* is responsible for education in certain fields of a particular science and provides appropriate sets of courses. For example, the Department of Computer Science and Engineering is responsible for computer science technologies and applications related to software. The goal of a department is to award degrees. In addition, the department sets goals concerning the content of the courses. It aims at developing the content of the degree programme as well as the content of single courses so that they are appealing to students and provide them with the skills and knowledge students need after graduation.

Planning phase from the organisation's point of view

The Rector and the Council of TKK decide how appropriations will be distributed between teaching and research. In addition, the Council and the rector decide what kind of additional measures of support are needed and how they are financed. The Council also to the plans the extent and subject matters of study entities that comprise basic studies (Figure 33, module P). However, the Council only gives broad guidelines concerning the content of the studies leaving the final decision to the degree programs.

The degree programs council plans the basic study entities (Figure 33, module O). Each faculty also constructs its degree programmes to best serve the needs of the industry and the field. The faculty makes strategic plans concerning the emphasis of the research fields. On the one hand, this ensures the standards the industry requires are maintained and, on the other hand, attracts good students who would graduate in time. On a practical level, the faculty co-ordinates lecture timetables and exams.

Each department makes plans how appropriations are best distributed to meet its goals. At the department level, the planning focuses on selecting the courses it offers to the students. The set of courses have to form entities that finally form a degree or part of a degree that the faculty has defined. Therefore, the planning consists of discussions on how entities of courses fit into larger degree curriculum and assuring that the supply of the courses is appropriate and meets the standards. On a longer time scale, the department also makes plans concerning the content of the degree programme and each course.

At the organisation level, the setting goals and planning phases are often closely tied to each other. One concrete venue for setting goals and planning is the performance negotiations that different levels of the teaching organisation have with each other: the Ministry of Education negotiates with the Council of TKK, the Council negotiates with the faculties and each faculty negotiates with its departments. These negotiations concentrate mainly on quantitative goals and allocation of appropriations. How many students each faculty is expected to graduate and how much money they get to do it with is determined by negotiation with the Council (Rector).

Implementing plans phase from the organisation's point of view

In the context of different levels of teaching organisation the *teaching/studying* phase is replaced with *implementing plans* phase since the organisation itself does not teach or study but provides the infrastructure and resources for teaching and studying. The Council of TKK makes sure that there are resources for teaching, for example, in the form of appropriations, premises and infrastructure. The university also indirectly produces services that further its goals for the whole university (e.g., library, computing centre, student services and pedagogical support for teachers). The faculty distributes appropriations further to the departments and makes changes to the content of degree programmes if needed. Faculties and departments also arrange tutoring as well as some orientation courses for the students to help them. Each department implements its instruction goals by providing a set of courses using the resources the faculty has granted them. Teachers give courses and are responsible for their content and implementation. The department's primary interest is not in individual courses but in the set of courses and the entirety they compose.

Processing outcomes phase from organisation's point of view

In the context of the teaching organisation, the *processing learning outcomes* phase is replaced with *processing outcomes* phase. This decision is based on the fact that an organisation has many different kinds of outcomes. For the teaching organisation, the number of degrees awarded is the outcome of the instructional process. The Council of the Helsinki University of Technology may, for instance, compare the number of the degrees the faculties have awarded during the previous year with the goals set earlier. The outcome of the instructional process for the faculty and the departments is the number of graduates, which reflects the overall success of the whole instructional process.

Feedback for the organisation

The different levels of an organisation receive feedback from several sources. One source of feedback is the organisation's own experiences from previous years. Organisation, as such, is an abstract concept. However, several people working at different levels of a university's administration may make notes on how the instructional process proceeds. The teaching organisation may also have generated procedures that it can use to detect development needs. For instance, each department keeps track of the number of students who enrol in its courses for resource management purposes. For example, if enrolment in a course is higher than expected, the department can allocate more resources to the course. In addition, the different levels of teaching organisations receive feedback concerning quantitative goals from each other, especially from the higher levels of organisation.

During the performance negotiations, the Council of TKK receives feedback from the Ministry of Education. For example, the longer-term trends concerning the annual number of graduates and the poor outcomes of a previous year may result in cutbacks to following years' appropriation. In a similar way, the faculties and departments receive feedback concerning the quantitative goals during the performance negotiations. The teaching organisation also receives quality-related feedback from society. The results of the audits of higher education institutions' quality assurance systems give feedback on the processes that support the attainment of the objectives of the institution. The selection of Centres of Excellence, on the other hand, gives feedback on the quality of the institution's operations and their results.

The third society-level feedback source is the statistics, surveys on how many graduates have been employed, and how the education the university has provided them has met the requirements of working life. Interviewing and sending questionnaires to alumni provides important feedback on the former instructional processes, which can be used as a basis for improving and updating the instructional process. Interviews and questionnaires provide feedback concerning the content of the degree programmes and the knowledge and ability levels of the graduates. For example, the study by McDonald and McDonald (1999) highlighted the following knowledge and abilities that computer science graduates should possess: a strong technical background in computer science, good communication skills, the ability to function effectively in teams and good problem-solving skills. The list was generated based on the opinions of faculty members, students, employers and alumni. This kind of study gives the organisation some indications of how they have succeeded in providing their instruction, teaching being a central part of that. Academic studies can also provide feedback that is needed in the setting goals phase. For example, Surakka's (2005) doctoral study reveals which content and skills are relevant for a graduating programmer entering professional life.

The faculties and departments might change the content-related goals according to the feedback industry, employees and alumni provides. For example, a faculty can decide to change a degree structure's emphasis if a need emerges for that. For example, in the 1990s, the faculty members of Helsinki University of technology (TKK) received requests and feedback that a new kind of professionals was needed who would be able to lead professionals of different fields and operate as a link between the end users and the developers of high technology. As a result, in 1999, the new degree programme, Information Networks, was launched. This interdisciplinary degree programme combines computer science, economics, and communication studies.

The teaching organisation can also collect feedback from teachers and students. These observations and the feedback collected can be used to develop the process. For instance, the Faculty collects student feedback from the courses departments offer to the students. The same course feedback can be used at both the faculty, department and course level. An organisation may also collect feedback after larger study modules and after graduation. Teachers may provide feedback to the teaching organisation, for instance, by stating that more resources are needed for some course.

International comparisons between universities provide also feedback to the university. The placement on such ranking lists gives some idea of how well the university has succeeded in producing and offering research, education and teaching of an international high standard. Another source of feedback is the students. The number and the quality of the students who apply to study reflect, in part, the respect for and the standard of the university. It is likely that, if the university has a reputation of producing qualified professionals who have good employment opportunities, then it also attracts applicants. The university also follows the average grades and the number of years it takes for students to graduate.

Yet, another feedback source for the university is the quality of its employees. A university that has a reputation of high-level research and teaching is also likely to attract the best researchers and teachers in the field. In addition, high-level research and research-based education are connected. If a university is able to employ the best professionals in the field, it affects the quality of the research that, in turn, may affect the quality of teaching.

Based on the feedback it receives, a teaching organisation may make some adjustments and changes to its processes. However, since the time cycle of the teaching organisation-level processes is long, from a year to several years, it is not always possible to utilise the feedback during the ongoing process. Instead, the feedback is utilised during the following processes.

To conclude

The feedback loop model that I have developed based on general system theory's (GST) tenets can be used to analyse the instructional process in a systematic way. The detailed analysis of the phases of the process and feedback loops provides the teaching organisation with the tools to analyse the strengths and weaknesses of its own actions and structures. The model can be used, for example, to analyse the feedback sources used, how the feedback is utilised and what are the consequences of the feedback. The feedback loop model is thus a tool that can be used in development work.

6 Research procedure and the quality of the empirical studies

This chapter describes the research procedures of the empirical part of this study. The empirical research was performed to gain knowledge of the concrete difficulties and challenges that the students, teachers and representatives of the administration face during the instructional process. The usability of the earlier developed analysis models was tested with the empirical data.

The three research questions focus on various aspects of the instructional process; in order to be able to answer these various questions, different kinds of research designs were needed. The first research question (*What kind of difficulties do students encounter when studying computer programming?)* was answered with a three-part design where quantitative and qualitative phases alternated or were used concurrently. A purely quantitative or qualitative approach would have provided only a limited answer to this question, therefore, I made the choice to use a mixed-method approach where both qualitative and quantitative methods are used to study the phenomenon (Tashakkori and Teddle 2003). Research questions two (*How do the CS teachers see the instructional process?*) and three (*How is the instructional process seen at the organisation level?*) were answered using the qualitative approaches.

Several different procedures were used during the study that illuminated the instructional process from different viewpoints. During the whole research process, both quantitative and qualitative approaches were used. Therefore, the research methods' philosophical premises differed. Quantitative research builds on positivism whereas qualitative research builds on postpositivism (Lincoln and Guba 1985). The former is based on assumptions of single reality, objectivity, time- and context-free generalisations, external law-like relations, and value-free research. The latter is based on assumptions of multiple realities, degrees of subjectivity, and context and value bound results. This study does not relate strongly to either of those traditions, but to pragmatism. In practice, this means that I have chosen the research approaches according to their suitability for the research questions. The research method's ability to answer the research question has been the criteria for choosing the method. In the following each research question and the procedures relating to it is discussed separately. The procedures are summarised in Table 7, Table 10 and Table 13.

The following paragraphs discuss the degree of validity and reliability of the research. Validity is a multi-faceted concept; the literature acknowledges several different types of validity. For example, Cohen, Manion & Morrison (2000) list 18 different types of validity. In addition, the criteria and interpretation of validity of the research varies depending on the research approach that is used. Qualitative research's premises differ from those of quantitative research and therefore they cannot always use the same criteria in a discussion of quality (Lincoln and Guba 1985). Next, I will introduce the concepts of validity and reliability that were used in this study.

Internal validity refers to how accurately the findings describe the phenomenon that is studied. More precisely "*Internal validity seeks to demonstrate that the explanation of a particular event, issue or set of data which a piece of research provides can actually be sustained by the data*" (Cohen et al. 2000, p. 107). In quantitative research, the level of internal validity depends on the quality of the data analysis and the interpretation of the results (Onwuegbuzie and Daniel 2003). The quality of the data analysis refers to the careful and appropriate use of statistical tests. This includes, for instance, the checking the underlying assumptions associated with a particular test. The quality of data interpretation refers to the proper interpretation of the results that the statistical packages produce.

In qualitative research, internal validity has to be assessed using different criteria. Cohen at al. (2000) suggest that the validity of a qualitative study can be assessed, for instance, through the richness and scope of the data that is achieved and the extent of triangulation. Lincoln and Guba (1985) use the concept of *trustworthiness* instead of internal validity. They set several criteria for the trustworthiness of qualitative research, for example, credibility and transferability. I will first focus on *credibility* and I will discuss transferability later in the next paragraph. The authors of the book suggest several techniques to achieve credible findings and interpretations. These include, for instance, prolonged engagement and peer debriefing. These techniques aim at familiarising the researcher with the context of the research so that she/he is able to understand better the interviewees or observed situations and to providing the researcher feedback on the analysis process.

External validity in quantitative research refers to the degree to which the results can be generalised (Cohen et al. 2000). Measures to meet the criterion of generalisation include randomised sampling to avoid biased data. Qualitative research, on the contrary, does not aim at generalisability of the results. Instead, it seeks to provide the reader the adequate information to judge whether the current results could be relevant in another place and time (Lincoln and Guba 1985). In other words, the *transferability* of the results depends on the degree of similarity between the current research context and the context the reader is thinking about. Therefore, it is not the author's, but reader's, task to estimate the degree of transferability. However, the author's task is to provide a profound description of the context, subjects, and the data analysis procedure.

Content validity refers to how the research instrument covers the items that it is supposed to cover (Cohen et al. 2000). For example, a questionnaire or an interview plan should include questions about all known relevant facets of the phenomenon. Content validity can therefore be interpreted in a similar way in both qualitative and quantitative research.

Construct validity relates to the operationalisation of the concepts that are used during the research process (Cohen et al. 2000). In other words, construct validity addresses whether the concepts describe the phenomenon at hand, and whether the researcher and the subjects understand them in the same way. In qualitative research, construct validity can also refer to the meaningfulness of the used categories for the participants and the degree to which the derived categories correspond to participants' experience.

These different aspects of validity refer to the ability of the research process to address the phenomenon it was aiming to address. *Reliability* has a slightly different focus – it concerns the quality of a research instrument, i.e. the consistency of the research instruments and measurements (Cohen et al. 2000). It can be understood as the *stability* of the instrument, meaning that a instrument will give similar results over time and over similar samples. Stability relates therefore both to the instrument and to the object of the research. Reliability may also refer to *equivalence,* which is the degree to which the equivalent forms of an instrument give similar results.

In qualitative research, the concept of reliability is controversial. The basic assumptions of reliability, that data and results should be replicable, contradict the assumptions of qualitative research. The premise of qualitative research is that the phenomenon and the situation are unique and therefore the research cannot be replicated exactly in the same way as it was done (Lincoln and Guba 1985). For example, people who are under investigation change as well as the circumstances around them. However, this does not mean that qualitative research should not observe aspects of accurate data collection and rigorous data analysis.

6.1 The procedure for studying what kind of difficulties students encounter when studying computer programming

The research concerning the students' difficulties in the introductory programming course (CS1) used a mixed method approach and had three parts (Table 7). It is characteristic for this approach to use both quantitative and qualitative data collection and analysis methods in one research project. A qualitative study might be used as relevant source of information for the following quantitative part. On the other hand, a qualitative study might be used to study further the aspects found in the earlier quantitative study. (Tashakkori and Teddle 2003)

The first part of the study took place in 2005 and included a quantitative survey of the experiences of the dropped out students in a CS1 course. The survey gathered feedback on the course and on the students' reasons for dropping out of the course. The motivation for the first part of the research project was to better understand the underlying factors that contribute to the non-major CS students' decisions to drop out and to highlight difficulties the students face during the instructional process. The non-major CS students were chosen as a target group because they clearly faced some difficulties during the course that contributed to the high drop-out rate. The results of the first part served as a starting point for planning the second part of the study. The second part consisted of a qualitative interview study, which focused on gathering thorough information on the nature of the reasons for dropping out. The results from the first part were used when making the interview plan. In a similar manner, the results of the second part were the base of the questionnaire that was sent out as the third part of the research project in 2006 and 2007. For example, the previously analysed reasons for dropping out were reformulated into questions where the respondents were able to estimate how much the reason at hand had affected their decision to drop out of the course. Before describing the research process in more detail, I will introduce the course from which the data for the first research question was gathered.

The course that was studied

This section describes the introductory programming course (CS1) from which the data for the first research question was gathered[8]. Students at a university of technology in Finland study a diverse range of technical courses. The basic studies module that the students have to take at the beginning of their studies includes several large introductory and advanced courses in subjects, such as mathematics, physics, and computer science. The courses that do not directly belong to the student's major aim at giving the student the ability to pursue their major studies, as well as providing some general knowledge concerning the technical sciences.

[8] I had taken this course myself in 2003 to gain a first hand experience on the course.

Table 7 Procedure addressing the research question 1

RQ1 What kind of difficulties students encounter when studying computer programming?			
Approach	QUANTITATIVE →	QUALITATIVE →	QUANTITATIVE + qualitative
Data collection method	Questionnaire	Interview	Questionnaire with some open questions
What was asked from the data?	Reasons for dropping out Feedback on CS1 course	Reasons for dropping out Experienced difficulties	Perceived programming related difficulties The distribution of reasons for dropping out Reasons for dropping out
Data analysis methods	Descriptive statistics Correlations Wilcoxon test Kruskal-Wallis test	Grounded theory Ad hoc approach	Descriptive statistics Wilcoxon test Mann-Whitney U test Correlation Factor analysis Free list
Target group	Dropped out students	Dropped out students	Dropped out students Students who passed the course
Which RQ was answered	1.1 Why do students decide to drop out of the CS1 course?	1.1 and 1.2 Which content-related issues do students find difficult?	1.1, 1.2 and 1.3 Are there statistically significant differences in what programming related issues students who drop out and students who pass the course find difficult? 1.4 What kind of strategies to overcome difficult issues do students find helpful?

Note. " →" stands for sequential, "+" stands for concurrent, capital letters denote high priority and lower letters denote lower priority.

As a part of their basic studies module, all students at Helsinki University of technology (TKK) take the course " Introduction to the computer environment" for newcomers. Most of the students take also a programming course in Java. The basics of computer programming are considered one of the compulsory skills that, for example, a Master of Science in Electrical Engineering should possess. The third course that many students take covers data structures and algorithms. The first three courses give the students the knowledge and skills to successfully continue CS studies if they so wish and to apply programming in their studies and future careers.

Helsinki University of Technology offers yearly two parallel introductory programming courses in Java (since the spring of 2009 the spring course teaches Python). The fall course is aimed at CS majors and non-major CS students who plan to take more CS courses. Meanwhile the spring course is aimed only for the non-major CS students. Typically, degree programs, such as Electrical Engineering and Mechanical Engineering are well represented in the spring course. Most students of the course are required to take a few computer science courses as a part of their basic studies module. The model plan for studies suggests that students should enrol in the introductory programming course on the spring term of their first or second year, not later. Yet, students can take the course whenever they choose. Approximately 500-600 students enter each spring and fall course every year. The drop-out rate at those courses has varied approximately 40-25% during the last four years. The main goal of the introductory programming course is to learn basic object-oriented programming in Java. The course uses an objects-early approach, introducing objects and classes in the second week. In the beginning, however, the emphasis is on basic procedural concepts, and design issues are not covered until the end of the course.

During the three years (2005-2007) when the data was collected, two different versions of the course were given. In 2005, the extent of the course was 8 ECTS, which corresponded 200 hours of work for the course. No previous knowledge of programming was required. The course was organised as follows. There were about 50 hours of optional lectures that covered most basic Java concepts (variables, types, objects, classes, program control, basic I/O, basic graphical user interfaces in Java, exceptions, interfaces, samples of Java collections). The lectures were given in Finnish. To pass the course, the students were required to solve 10 sets of weekly exercises that contain several small program implementation exercises. In addition, students had to carry out a personal programming project, where a larger program (some 300-500 LOC, or more) was implemented, and then take a written examination.

An automatic assessment tool was used to grade the submissions of the weekly exercises and give feedback on them to the students. Several resubmissions were allowed for each exercise. Achieving 50% of the maximum points was enough to pass the exercise part of the course, but gaining more points improved the final grade of the course. Students had an opportunity to also attend the closed labs, organised several times per week, where they were able to ask for help from an assistant teacher (typically a 2^{nd}–4^{th} year student) if they were having difficulties with the programming exercises. The programming project and the examination were graded manually, and they affected the final grade.

In 2006 and 2007, the course was split into two parts. After the split, the extent of the first course was five ECTS, which corresponds to proximately 140 hours of work. The organisation of the first course was otherwise the same as the old course but no personal project work was required. In 2006 and 2007, 25–35 hours of optional lectures were given during the course (a listing of the content of the lectures in 2007 are in Table 8). In both course versions the students were able to use textbooks[9], educational material provided by the teacher[10], course slides, the course newsgroup and the Internet (such as the extensive web site of the course) as their learning resources. The lecturer was also available during his/her office hour. There were guidelines concerning the students' collaboration stating that discussing about the code with each other is OK but everybody has to write his/her own programs. Control actions were taken to curb plagiarism.

[9] The official textbook that was used in the course was in English. However, students were free to use any textbook as a source of information.
[10] A Finnish extended handout about the course content written by the teacher.

Table 8 The content of the lectures and programming exercises in 2007

	Topic of the lectures and programming exercises
1. week	Variables, assignments, output on screen, expressions, if statement, Scanner class
2. week	Class, object
3. week	More about objects, programming style, loops
4. week	Objects as instance variables, overloading, characters, strings and arrays, searching arrays
5. week	More about arrays, ArrayList
6. week	Sorting an array, class variables and methods, recursion, formatted output of decimal numbers, linked list
7. week	Inheritance, abstract classes, interfaces
8. week	Exceptions, reading input, file I/O
9. week	Handling binary files, designing classes

The research procedure

Part I: Quantitative survey of student who dropped out of the CS1 course

The first part of the research project aimed at answering the question *1.1 Why do students decide to drop out of the CS1 course?* The impetus for this first survey was the alarming learning outcomes from the introductory programming course: the course's drop-out rate was high. This was a sign for the teaching personnel that something was not working well during the instructional process. Therefore, a questionnaire[11] was designed to explore the possible reasons for the high drop-out rate. The course feedback form and the feedback students had given using the form, were used when the questions and the answering options were designed. However, typically only students who pass the course fill in the course feedback form. Therefore, the answers to the feedback form were able to give only some preliminary ideas of the drop-out reasons.

The existing literature also guided the composition of the questionnaire. However, the literature reviews revealed only three studies that discussed the students' reasons for dropping out of an introductory CS course (Meisalo, Suhonen et al. 2002; Meisalo, Sutinen et al. 2002; Xenos et al. 2002). Time management was the only reason for dropping out that came up in all three studies. The other reasons included, for example, too difficult exercises, the students' unrealistic preconceptions of the level of the difficulty of the course, and lack of assistance from the tutor. The third source for questions was anecdotal information concerning the difficulties students confront in the course. The anecdotal information was based on informal individual discussions with the course teachers. These three sources, course feedback, literature, and the experiences of the course teachers, were used to make sure the questionnaire covered the essential aspects of the phenomenon and thus enhanced the content validity of the research. The course feedback form was also helpful when designing the questions. The questions that were used in that form and the answers students had given to them gave some guideline as to how students generally understood the questions.

[11] All questionnaires were in Finnish

The web-based questionnaire aimed at mapping the possible reasons for dropping out since it was not well known why so many students at this particular course dropped out. The questions were chosen so that they would give some information about whether there was something in the course organisation that affected the decision to drop out. The questionnaire included both closed-ended and open-ended questions concerning course content, study material, teaching, perceived difficulties and drop-out reasons. The open-ended questions were mostly ”Other reason, please elaborate” type questions. The last question in the questionnaire inquired about the students' willingness to come to an interview to talk about the course and the reasons that lead to their decision to drop out of the course.

At the end of the course, in late spring 2005, an email message containing a request to participate in the research and a link to the questionnaire (Appendix 2) was sent to students who had signed up for to the CS1 course at the beginning of spring 2005 term but had failed to finish the programming exercises and/or the project work. To build on students' motivation to answer the questionnaire, the email that contained the link to the questionnaire included also a description of the ongoing research and a comment that this questionnaire gives the dropped out students the chance to give feedback and to influence future courses.

The first reminder was sent a couple of weeks later and the second and the third reminders were sent at the beginning of the fall semester. The final response rate was 49.5%. Even though the response rate was not as high as in some other surveys reported in the field it is still fair taken that the topic of the questionnaire was somewhat embarrassing or at least it asked about a negative incident (dropping out of the course).

The statistical analyses were performed using StatView and SPSS 16.0. A chi-square test was run to determine how well the respondents represented all the drop-outs in the course. A Kolmogorov-Smirnow test was used to determine whether the data was normally distributed. Based on the results the nonparametric Wilkoc and Kruskal-Wallis tests were used when determining the statistical differences. In many places, descriptive statistics were used also.

Finally, the results of the research were written in a conference paper, which was submitted to peer review to ensure the quality of the data analysis and the interpretation of the results. Discussions with experienced researches at conferences (ICER, Koli Calling) provided valuable feedback on the quality of the results and future research plans. For example, the feedback confirmed that a qualitative interview study would provide a valuable inside to the drop-out phenomenon.

Part II: Qualitative interview of sample of students who dropped out

The second part of the research was a continuation of the first part, which had provided an overview of students' reasons for dropping out of the course. However, that survey provided information only on those aspects that were asked and the closed-ended questions did not give the students the opportunity to tell their interpretation of what happened. Therefore, the aim of the second part was to get profound information on the characteristics of drop-out reasons and thus answer the questions 1.1 *Why do students decide to drop out of the CS1 course?* and 1.2 *Which content-related issues do students find difficult?* In the questionnaire that drop-outs filled out, there was a question whether the respondent would be interested in coming for an interview. The respondents were encouraged to volunteer for the interview by explaining how important it is for the university and teaching personnel to receive first-hand feedback from students – especially the ones that are struggling with their studies. In addition, it was made clear that the interviewer was an outsider (did not belong to the teaching staff) who would keep the information confidential and anonymous when reporting the results. No incentive, except a cup of coffee, was offered for volunteers due to the restricted resources. Eighteen students who had expressed their willingness in the questionnaire were interviewed during fall term 2005, and in the beginning of the year 2006.

The interview plan was based on literature and the results of the questionnaire. The results of the survey informed the interviewer about possible relevant reasons to drop out of the course. The interview plan was designed so that it would help the researcher to get information that is more profound on the earlier found reasons. The plan was also discussed with other researchers in the field who had experience of doing computing education research. The type of interview used was methodologically between an informal conversational interview and an interview guide approach (Johnson and Christensen 2004). This means that there was a loose interview plan (Appendix 3) that guided the conversation. Depending on interviewee's personality and "talkativeness", the interview followed the one or the other method more closely. However, in both cases the most essential topics were covered. In addition to questions concerning the CS1 course itself, social aspect related questions were included in the interview plan. This decision was based on literature and the results of the survey, which indicated that the reasons to drop out are manifold. The time used for interviews varied between 35–85 minutes with the average close to 50 minutes. The interviews were held in Finnish, tape-recorded, and transcribed by the author. In addition, notes where taken manually during the interviews. In four cases, tape recording was not possible because interviewees did not consent or because of technical difficulties; in these cases only notes were taken. Discussions with experienced researchers in the field ensured that the language of the interview data did not have an impact on the data analysis process. The structure and the vocabulary of the Finnish language are such that there are no big differences in data analysis procedures compared to the situation where the data is, for instance, in English.

In an interview research the researcher is a data collection instrument her/himself. Therefore, it is important that the instrument is able to detect the essential aspects of the interviewees talk to guide the discussion into a fertile direction and to be able to analyse the data. The literature suggests prolonged engagement to enhance the trustworthiness of qualitative research (Lincoln and Guba 1985). I had worked at the department that organises the CS1 course for over three years before this study took place. During that time, through informal discussions with course teachers, I became familiar with the challenges relating to the CS1 course. In addition, a few years before this study took place I, the interviewer, had taken the same introductory programming course and one other computer science course. Therefore, I had first hand experience and knowledge of the course content and the challenges students face during the course. To build trust I mentioned at the beginning of the interview that I had recently taken the same course and thus had first hand experience of the challenges students face during the course.

Grounded theory was used to analyse the data (Glaser and Strauss 1967; Ryan and Bernard 2003; Strauss and Corbin 1990). I chose grounded theory as the data analysis method because it gave the opportunity to approach the data with an open mind. The literature review had revealed only few reasons for dropping out along several factors that may cause difficulties to students. On the other hand, the results of the first quantitative part of the research gave a thin impression compared to the rich interview data and thus did not provide relevant categories for data analysis. After the first reading of the interview data it was clear that time management was a repeating theme that came across in the literature review, in the quantitative part of the research and finally also in the qualitative interview data. However, the initial reading did not reveal enough relevant similarities between the findings of the literature review and the interview data to use the categories found from the literature as a base for the analysis categories. Moreover, the interview data appeared to reveal something about the influencing factors behind the final reasons for dropping out of the course. For example, the interview data contained factors contributing to the time-management difficulties. The literature review was not able to highlight relevant categories to address these possible causal relations. Based on these initial notions on the data, I chose grounded theory for data analysis method.

The analysis process started with reading the transcripts and the notes taken during the interviews through several times. Furthermore, I read the interviewed students' answers to the questionnaire in order to receive as much information on the student as possible. During the first readings of the transcripts open and axial coding intertwined (Strauss and Corbin 1990). The data was read sentence by sentence and in each statement concerning reasons to drop out, I identified and labelled the reasons mentioned. As several emerged, I started to categorise them according to their similarities. Statements with similar reasons were combined, and a more general title for each category was given. As a result, a number of categories emerged describing qualitatively different reasons for dropping out of the course.

During the axial coding, I started to look for possible connections between the categories. The connection building was done by looking at the conditions and consequences of the different categories and then making deductions about what the reasons were behind the decision to drop out of the course and what the relation was between the reasons. Each deduction was checked against the original interview data. Finally, I drew drop-out reason networks for each of the 18 interviewed students to get an overview of the series of events that had led to the student dropping out of the course. The networks were then compared with the answers students had given when filling in the questionnaire. I gathered the observations on that comparison into the tables (see Appendix 8), which collected up the students' stories including their previous experiences concerning programming, plans, experiences during the course, and finally the reasons for dropping out. The last part of the analysis is best described as an ad hoc approach to analysis where several techniques are used to derive meaning in rich data (Kvale 1996).

During the data analysis phase, I discussed the results and the procedure with other researchers in the field. These researchers had computer science education backgrounds and they had several years experience of teaching programming courses. Thus, they had first hand knowledge of the difficulties students confront in the course. With this background, they were able to give valid comments concerning the procedure and the results. Finally, the results of the research were reported in a conference paper, which was submitted to the peer review to ensure the quality of the data analysis and the interpretation of the results. Feedback that was received at the conference was also valuable in evaluating the meaningfulness of the results for the international community of computer science teachers and researcher.

Part III: Mainly quantitative survey of both dropped out and passed students

The aim of the third part of the research procedure was to discover how common the earlier discovered reasons for dropping out were. In addition, the programming related difficulties were investigated in a more detailed manner. During this time, also the students who passed the CS1 course were included into the target group. This decision was based on the desire to get a more comprehensive picture of what students found difficult in the course and to highlight the difference in programming related difficulties that made some students to drop out. The third part of the research project was aimed at answering the questions *1.1 Why do students decide to drop out of the CS1 course?, 1.2 Which content-related issues do students find difficult?, 1.3 Are there statistically significant differences in what programming related issues students who drop out and students who pass the course find difficult?,* and *1.4 What kind of strategies to overcome difficult issues do students find helpful?*

Two questionnaires were created; one was aimed at the students who passed the programming exercises and the other was aimed at the students who failed to get a grade from the exercises (Appendix 4). Based on the experiences of a course teacher, the written examination has not been the critical issue when it comes to passing the course; the programming exercises were struggle for the students. The students who got a good grade from the programming exercises usually had gained such knowledge and skills that they had little difficulties with the exam. Therefore, in this part of the study the students who got a grade from the exercises are treated as if they would have already passed the whole course.

The questionnaire was based on the results from the quantitative and qualitative parts described above. Especially the results from the interview study provided a fertile base for the planning of the questionnaire. The aim of the questionnaire was to discover how common the earlier analysed reasons for dropping out were and to investigate the difficulties that students confront during the CS1 course. The first parts of the questionnaires were identical; they contained some background questions: questions concerning students' plans, study motivation and studying skills. The next part contained questions that mapped how difficult it was for the student to learn certain programming related skills and knowledge. The third part of the questionnaire was given only to the students who dropped out. This part aimed at identifying the reasons for dropping out. Based on the earlier performed qualitative survey, the possible reasons for dropping out were listed, and respondents were asked to estimate how much each of the listed reasons affected their decision to drop out. In addition, open-ended questions were added in case the respondent did not find suitable reasons in the list.

At the end of the course in 2006, both questionnaires were sent and three reminders were sent to those who did not respond. The final response rate for drop-outs remained low (25.9%) even after sending out the reminders. This was somewhat surprising since the timing and the procedure of sending out the questionnaire was similar to the previous year when the response rate reached nearly 50%. Research methods literature suggests incentives to enhance the response rate. Due to the restricted resources, it was not possible to offer respondents any money value incentives. Sometimes extra course points are given to students as an incentive for answering surveys. However, since the target group that had a low response rate was the drop-outs such an incentive would have been meaningless for them. Instead of offering incentives, I built up the drop-outs motivation to answer the questionnaire by explaining that this is their chance to give feedback and to have an impact on future courses. The results of Chi-test suggest that gender, year of study, and degree program of students who filled out the drop-outs questionnaire in 2006 were not statistically significantly different from all drop-outs in the same course. However, because of the low response rate there is a danger that the data might be biased and thus the results of this survey must be interpreted with this notion in mind.

In 2007 the questionnaire for the students who passed was sent at the end of the course, as was done in the previous year. However, this time the questionnaire for the dropped out students was sent in the middle of the course to those students who, based on their poor performance at the course, had no possibility of passing the course. This change in the timetable was made to enhance the dropped out students' response rate. As a result, in 2007, the drop-outs' response rate increased to 45.5%. The topic of the survey may have had an affect on the response rates. Elaborating on a negative incident, such as dropping out of a course, might have been somewhat sensitive for some students. A few drop-outs even took the time to send an email to the researcher stating that they will not answer the questionnaire. The results of the survey must be interpreted keeping the response rate in mind.

The response rates for the passed students' surveys were 47.5% in 2006 and 52.6% in 2007. Students were also asked to fill in a standard end of the course feedback survey at the end of the spring term. The fact that there were two surveys to be filled in almost at the same time may have affected the passed students motivation to answer the surveys.

The statistical analyses were performed using StatView and SPSS 16.0. Chi-square tests were used to determine to which degree the respondents represented all passed and dropped out students in the course. The Kolmogorov-Smirnov test was used to test whether the data was normally distributed and appropriate tests were chosen based on the results. Non-parametric versions of the t-tests, the Wilcoxon and the Mann-Whitney U tests were used to determine whether the groups had statistically significant differences. In addition, descriptive statistics were used.

After retrieving the mean values of each reason for dropping out I used explorative factor analysis to compress the 22 reasons. Before the actual factor analysis, I made pre-analysis to examine whether the data can be used for factor analysis. The general rule of thumb is that the subject to variable ratio should be at least 2:1, and the ratio of 5:1 is preferable (Onwuegbuzie and Daniel 2003). In this study, the ratio was > 5:1. The number of subjects (119) is a modest sample size for a factor analysis and therefore the results should be interpreted with care.

The result of Bartlett's test demonstrated that the inter-item correlations were sufficient (χ^2 (231) = 729.61 $p < 0.001$). In addition, the result of the Kaiser-Meyer-Olkin test (KMO) of sampling adequacy also suggested that the data could be used for factor analysis. The KMO value was 0.58, which is above the 0.5 criteria.

Since the data was not normally distributed in all variables (most likely due to the small sample) I used the Generalized least squares factor analysis method. Finally, I used the Varimax rotation to produce a simple and meaningful factor structure. The variables' communality varied between 0.32 – 0.71, the mean being 0.50. There was no reason to drop any variable out of the analysis since all h^2 values were > 0.20.

The analysis produced seven factors that had Eigenvalue greater than one. However, the last two factors explained little of the variance. In addition, the scree plot suggested that five factors could be possible. The last and the most important criterion for choosing five factors over seven or six was the possibility of interpreting the factors. Five factors provided a structure that was informative, meaningful, and clear.

The internal consistency of the factors was calculated with Cronbach's alpha. High alpha value indicates that items loading to the same factor measure the same concept. The alpha for the first factor was 0.83, for the second factor 0.73, for the third factor 0.40, for the fourth factor 0.65, and for the fifth factor 0.71. The internal consistency of four factors was fair, but factor three, *Time managing and preferences* was doubtful ($\alpha = 0.40$). The correlation between the factors showed that the factors did not correlate with each other. The highest correlation found between two factors was 0.015.

The answers to the open ended questions were usually short; often the answers comprised of a few words or short sentences. I used the free list –method to analyse the answers (Ryan and Bernard 2003). For example, the answers to the question *What got you through the difficult issues* produced a list of strategies, which were then divided into categories according to their similarities. Finally, I counted the frequency of the mentions in different categories.

Finally, the results were reported in a research paper, which was submitted to a peer review and presented at a conference. This process provided feedback on the quality of the research process and the meaningfulness of the results. For instance, according to the feedback the computer science teachers agreed with the idea that there often are several concurrent reasons for dropping out a course. Table 9 summarises the quality enhancing measures that were taken during this three-part research on the student

Table 9 Summary of measures taken to enhance the quality of the research on students in the introductory programming course (CS1)

RQ1 What kind of difficulties students encounter when studying computer programming?			
	Part I: QUANTITATIVE	Part II: QUALITATIVE	Part III: QUANTITATIVE + qualitative
Before data collection	Content validity: course feedback, literature, anecdotal information from the teachers Construct validity: feedback form as a base for the questionnaire to ensure understandable questions	Content validity: literature, results of the previous quantitative phase of the study Construct validity: Discussion on interview plan with other researchers Trustworthiness: prolonged engagement Building on motivations of the respondents	Content validity: literature, results of the previous phases of the study Construct validity: using the experiences of the previous parts of the study to formulate the questions
During data collection	Building on motivations of the respondents Reminders to enhance the response rate	Construct validity: clarifications of the used concepts during the interview if needed Recording interviews and taking notes during the interviews	Building on motivations of the respondents Reminders to enhance the response rate
Data analysis & conclusions	External validity: Chi-square to determine how well respondents represented all drop-outs Internal validity: test to explore is normality assumption violated and choosing the appropriate tests according to the results Feedback from other researchers and CS teachers	Transferability: profound description (context, data analysis, text extracts) so that reader can determine whether transferability is possible Trustworthiness, credibility: peer debriefing Using all the data Feedback from other researchers and CS teachers	External validity: Chi-square, how well sample reflects the population Internal validity: test to explore if normality assumption is violated and choosing the appropriate tests according to the results Feedback from other researchers and CS teachers

6.2 The procedure for studying how the computer science teachers see the instructional process

The second research question (*How do the CS teachers see the instructional process?*) was tackled with a two-part research project (Table 10). The first part aimed at discovering what computer science teachers think about the students' studying process and their own possibilities to affect it. The second part focused on what the teachers think about the teaching process. The results relating the second research questions are reported in chapter eight.

Part I: Qualitative survey

The first part of this research project tackles the following four research questions: *2.1 How do CS teachers define what studying is? and 2.2 Which aspects of studying do CS teachers think students have difficulties with? and 2.3 Which aspects of studying do CS teachers think are focal for successful studying?, and 2.4 How do CS teachers think they can affect the students' studying process?* The data was gathered using a questionnaire (Appendix 5) that contained first some background questions and then open-ended questions concerning how teachers perceive what studying is, what are focal aspects of studying that contribute to successful studying, what are aspects of studying that teachers think students have difficulties with, and how teachers think they can affect students' studying process. The literature review discovered some studies on teachers' conceptions. However, the focus of those studies was more on how teachers perceive learning and not so much on what studying is. The study by Bruce and Gerber (1995) touches upon studying by reporting on teachers' conceptions on how learning is achieved. This study differs from that of Bruce and Gerber by the emphasis on its more general definition of studying. Whereas the study by Bruce and Gerber reported on how teachers think learning is achieved, this study asks teachers to define studying.

Table 10 Procedure addressing the research question 2

RQ2 How do computer science teachers see the instructional process?		
Approach	QUALITATIVE →	QUALITATIVE
Data collection method	Questionnaire with many open questions	Interview
What was asked from the data?	Teachers' conceptions of studying and teaching	How do teachers see the instructional process?
Data analysis methods	Grounded theory	Qualitative content analysis
Target group	CS teachers	CS teachers
Which RQ was answered	2.1 How do CS teachers define what studying is? 2.2 Which aspects of studying do CS teachers think students have difficulties with? 2.3 Which aspects of studying do CS teachers think are focal for successful studying? 2.4 How do CS teachers think they can affect the students' studying process?	2.5 How do the computer science teachers consider the different phases of the instructional process?

The data was collected with a web-based questionnaire. The other option to collect the data would have been interviewing computer science teachers. Interview method has several advantages: for instance, interviewer can define questions that interviewee finds unclear and interviewer can encourage interviewee to elaborate more on his/her thoughts if interviewee otherwise tends to give short, uninformative answers. However, interviewing is a time consuming method (time spend on interviewing and transcribing). The advantages of the questionnaire include aspects, such as it encourages honesty (especially if answers are anonymous) and it is more economical than the interview. (Cohen et al. 2000) The decision between questionnaire and interviews was made based on three reasons. First, since the target group was CS teachers I assumed they would be able to elaborate on their conceptions of studying in writing. Second, a web-based questionnaire was an effective way to receive answers from many teachers. Third, a questionnaire was also cost and time effective way to collect the data compared to travelling around Finland to meet teachers and transcribing recorded tapes. In this study I chose to use questionnaires as a data collection method. The open ended questions in the questionnaire were such that if I had used the interviews as a data collection method I could have included the same questions in my interview plan. Therefore, in content wise there were no big differences between the two possible data collection methods.

The questionnaire was aimed at teachers who had first hand experience of teaching computer science at university level. In fall 2006, the aim and the background of the study were briefly introduced at a national small-scale seminar that was aimed at teachers and researchers interested in computing education. The participants of the seminar were chosen as a target group because they represented well the different Finnish universities. A week later, the 41 participants of the seminar were approached with an email that contained a link to the questionnaire. Four reminders were sent to the ones that did not respond. Only the responses from persons who had experience of teaching and/or tutoring were taken into account in this analysis. Twenty-five teachers and assistant teachers answered the questionnaire and thus the final response rate was 61.0%. In general, the CS teachers' answers to the open ended questions were several sentences long. The most respondents elaborated on their conceptions and gave examples of their actions.

The original target group of the survey may have caused some biases to the data. Teachers who voluntary participated in the seminar might be the ones who were initially more interested in education and in how to improve their teaching than an average CS teacher. Thus, the responses might represent more sophisticated ideas than what would be received from teachers who are not particularly interested in developing their courses.

Grounded theory was used to analyse the written answers to the first two open ended questions (Glaser and Strauss 1967; Ryan and Bernard 2003; Strauss and Corbin 1990). I chose grounded theory as a data analysis method because the literature review did not provide reasonable predefined categories for the analysis of this data. The process of analysis started with reading through the responses to the two first open questions several times. Also open coding took place during these first readings. After a while, differences in focus[12] between the answers started to emerge. After the various focuses were identified, they were given a label. The axial coding took place partly simultaneously with the open coding. At the same time as I labelled the found focuses I also started to group similar focuses into emerging categories. This procedure resulted in main categories and sub-categories.

I used a slightly different approach to analyse the three remaining open ended questions in the survey that considered *teachers' conceptions of essential aspects of successful studying, aspects of studying that are difficult for students and how teachers can affect the studying process*. I used the categories and sub-categories that the analysis of the answers to the question *What kind of aspects are involved in studying?* had produced, as a starting point for the analysis. The analysis started with the reading and re-reading of the written answers. I labelled the emerging themes and then placed them into the existing categories. There was no need to create new categories thus implicating that the original categories were valid in describing teachers' conceptions. However, the properties of categories grew more diverse the further the analysis proceeded, since additional data revealed new properties.

[12] The question concerned what studying is and aspects of studying. Typically, respondents' answers focused on some particular aspect of studying, e.g., the student as an actor. Focus refers here to those aspects that the respondents emphasised in their answers.

One of the ways qualitative research aims at enhancing the quality of the research is prolonged engagement. This should help the researcher to become “tuned” to understand the relevant aspects of the phenomenon. Before designing this questionnaire, I had been working in the field of computer science education for several years with other researchers and teachers. Several of my colleagues were experienced computer science teachers. Discussions and joint research projects had taught me to better understand the world of computer science and what it is like to be a teacher in this field. To enhance the validity of the research I discussed about the procedure and the results with other researcher in the field of computing and computer science education. These discussions provided valuable feedback at different phases of the research process.

Part II: Qualitative interview

The second part of this research project focused on how the teachers saw their own teaching process. This part of the study aimed at answering research question *2.5. How do the computer science teachers consider the different phases of the instructional process?* The interviewees were chosen based on the course they taught. The teachers who were involved with an introductory programming course or the second programming course were a natural choice to keep consistency with the data collected from the students in an introductory programming course (CS1). Qualitative interview was chosen as the data collection method to gain an insight on the teachers’ teaching process. The aim of the interview was, on the one hand, to highlight at least some aspects of the reality that set boundaries for teachers’ actions and, on the other hand, to bring out the possible feedback sources and feedback loops in the teaching process. I used the feedback loop model that I had developed for this study (see section 5.2) as a starting point for the development of the interview plan (Appendix 6). The type of interview conducted was the interview guide approach (Quinn 2002). Topics, such as the different phases of the teaching process (setting goals, planning, teaching, outcomes), the feedback sources, and the utilisation of the feedback were covered in each interview, although the order and the wording of the questions varied from one interview to the next.

The interviews were held in the spring of 2008. The time used for each interview varied from 50 to 80 minutes, with an average time being just over an hour. All four interviews were held in Finnish. The interviews were tape-recorded and transcribed by the researcher. In one case, a short follow-up interview was held to discover the outcomes of a teaching experiment that were not yet known at the time of the first interview. The aim of the interview was to receive an insight into computer science teachers’ way of perceiving the instructional process. The high degree of generalisability of the results was not the goal of this research. Instead, I aimed at understanding profoundly the way some CS teachers experienced the instructional process. From this point of view, four interviewees were enough to provide the information that was needed for this study.

In the analysis phase I applied the qualitative content analysis method (Mayring 2000). The feedback loop model I had developed for this study provided a framework for the analysis. I used the deductive category application where the phases in the instructional process model were used as preliminary categories into which the quotes from the data were placed. The phases of the instructional process (setting goals, planning, teaching/studying, processing learning outcomes) became the main categories. The type of feedback received at each phase and the adjusting measures that were taken became the subcategories that were used in this analysis. An example of category definition and a coding rule is Table 11. Since I was the only one involved in this research project, it was not possible to check the reliability of the coding, for instance, with comparing the categorisation results of several trained researchers. The data was analysed sentence by sentence by placing the text extract into the appropriate category. As a result, there emerged a description on what different phases of the instructional process mean for the computer science teachers. Table 12 summarises the measures I took to enhance the quality of this research project.

Table 11 An example of a coding rule

Category	Definition	Example	Coding Rule
Main category 2: Planning	Discusses: • How the goals of the course will be achieved • Practical arrangements of the course, e.g., timetable of the course, hiring assistant teachers • Aspects that teacher has to take into account when planning	*… the most important aspect of it is to plan the programming exercises so that they are related to the goals and, of course, other teaching methods, lectures and materials, too. Exam should relate to the goals too…*	The extract discusses one or several aspects of the planning phase.

Table 12 Summary of measures taken to enhance the quality of the research on how teachers see the instructional process

RQ2 How the computer science teachers see the instructional process?		
	Part I: QUALITATIVE	Part II: QUALITATIVE
Before data collection	Content validity: literature Trustworthiness: prolonged engagement Building on the motivation of the respondents, clarification of the purpose and the goal of the research to the respondents before sending out the questionnaire	Content validity: literature, the development of the feedback loop model Construct validity: using the feedback loop model to formulate the questions, dividing the model into several smaller parts to retrieve understandable questions Trustworthiness: prolonged engagement
During data collection	Building on the motivation of the respondents	Construct validity: clarifications of the used concepts during the interview if needed Recording interviews and taking notes during the interviews
Data analysis & conclusions	Transferability: profound description (context, data analysis, text extracts) so that the reader can determine whether transferability is possible Trustworthiness: prolonged engagement, feedback from other researchers	Transferability: description of the context, procedure, interviewees, many extracts from the interview data Trustworthiness: prolonged engagement

6.3 The procedure for studying how the instructional process is seen at the teaching organisation level

The third research question (*How is the instructional process seen at the organisation level?)* was approached with a two part qualitative research (Table 13). The first part of the research focused on how the phases of the instructional process are represented in formal documents such as acts, and mission related statements of the university. The second part of the research focused on how the representatives of the administration see the instructional process. One of the goals in both studies was to highlight the type of feedback that was available to the administration, while the other was to discover the aspects that were perceived as challenging by them.

Table 13 Procedure concerning the research question 3

RQ3 How is the instructional process seen at the organisation level?		
Approach	QUALITATIVE →	QUALITATIVE
Data collection method	Selection of formal documents from the different levels of the teaching organisation	Interview
What was asked from the data?	How different phases of the instructional process are represented in documents	How do representatives of the administration see the instructional process
Data analysis methods	Qualitative content analysis	Qualitative content analysis
Data source	Documents (acts, mission statements, study guide)	Representatives of the administration and employees with administration related tasks
Which RQ was answered	3.1 How are the different phases of the instructional process and the feedback about the process and outcomes presented in documents?	3.2 How do representatives of the administration see the instructional process?

Part I: Qualitative content analysis

The first part of the research answered the question *3.1 How are the different phases of the instructional process and the feedback about the process presented in documents?* The documents for this part of the study were chosen based on two requirements. First, the documents stated something about the instructional process that considers the basic studies or the quality of teaching and education. Second, the chosen documents represented the different organisation levels: society, university, and faculty and department. The goal of the analysis of the documents was to receive an overview on those written guidelines that steer the teachers' and administrative personnel's' work. In this sense, the analysis of the documents was also an important preparatory work for the next part of this study, which contained the interviews of the administrative personnel.

I applied the qualitative content analysis method (Mayring 2000) in the data analysis phase. The earlier developed feedback loop model (section 5.2) provided a meaningful framework for the analysis. The four phases of the instructional process (setting goals, planning, teaching/studying, and processing learning outcomes) were slightly altered so that they better reflected the teaching organisation's processes. The main categories that were used in the analysis were: goals, plans, implementing plans, and outcomes.

The data analysis started with reading the documents and highlighting every sentence that concerned goals, plans, implementation of the plans, or the outcomes. There were two rounds of analysis. During the first round, the focus was on the different topics that were expressed in the different level documents and how the continuum from goals to plans and further to the implementation of plans and finally outcomes was expressed.

During the second round of analysis, the focus was shifted to how the themes found in the different level documents were presented in other level documents. Finally, the main instructional process related statements were summarised to give the reader a general overview of the content of the documents. The tables in chapter 9 provide examples of how the text extracts were categorised.

Part II: Qualitative interview

The second part of this research procedure answered the question *3.2 How do representatives of the administration see the instructional process?* I collected the data for the second part of the study by interviewing three representatives of the administration during the academic year 2007/2008. The interviewees were selected so that they represented slightly different viewpoints to the organisation. One was a professor whose position is somewhat between the teachers and the administrative personnel, one was a planning officer who looked at the instructional processes purely from the administrative point of view. The third interviewee was a person who had a broad view of the university because of her work history and current position. Again, I used the feedback loop model (section 5.2) as a base for the interview plan (Appendix 7). The type of the interview was the interview guide approach (Quinn 2002). The time used for each interview varied from 45–65 minutes. All interviews were held in Finnish and they were tape recorded and transcribed by the researcher. In addition, notes where taken manually during the interviews. Topics of the interviews included the different phases of the teaching process, the possible feedback sources, and how the feedback was utilised.

The generalisability of the results was not the goal of this research. I aimed at understanding profoundly the way some representatives of administration experienced the instructional process. From this point of view, three interviewees, who represented three different viewpoints to administration, were enough to provide the description that was needed for this study.

I used qualitative content analysis in the data analysis phase (Mayring 2000). The phases of the instructional process provided the main categories (setting goals, planning, implementing plans, outcomes). The data was also read through with special attention to the difficulties and problems the interviewees reported concerning the instructional process.

The aim of this research procedure was to explore how the instructional process is seen from the teaching organisation's point of view. I took the following measures during this research project to enhance the quality of the process. First, the results of the analysis of the documents gave general information on aspects that steer administrative personnel's' work and thus helped me, as an interviewer and analyser, to ask relevant questions and better understand what interviewees were talking about. Second, the development of the feedback loop model provided a framework for the interview plan and enabled the systematic analysis of the interview data.

The goal was to describe the process and highlight some problems that could benefit from further investigation and interventions. Therefore, generalisability of the results was not a goal for this research. Instead, I have provided descriptions of the data, the analysis process and the results so that the reader may estimate the validity of the results in his/her own context. Table 14 summarises the measures that were taken to ensure the quality of this research project.

Table 14 Summary of the measures taken to enhance the quality of the research on how the instructional process is seen at the organisation level

RQ3 How is the instructional process seen at the organisation level?		
	Part I: QUALITATIVE	Part II: QUALITATIVE
Before data collection	Content validity: the development of the feedback loop model	Content validity: the development of the feedback loop model, the results of the previous part of the study Construct validity: using the feedback loop model to formulate the questions, dividing the model into several smaller parts to retrieve understandable questions
During data collection	Including all relevant documents into analysis	Construct validity: clarifications of the used concepts during the interview if needed Recording interviews and taking notes during the interviews
Data analysis & conclusions	Transferability: descriptions of the analysis process, reporting the categorised text extracts in chapter 8. Making the categorisation explicit.	Transferability: description of the interviewees' background, extracts from the interview data

7 The instructional process from students' point of view

This chapter is the first of the three chapters that presents the empirical results of the research. This chapter focuses on the challenges and the difficulties students encounter when they study computer programming. The more precise research questions that this chapter answers are: *why do students decide to drop out of the CS1 course, which content-related issues do students find difficult, are there statistical significant differences in what programming related issues students who drop out and students who pass the course find difficult,* and *what kind of strategies to overcome difficult issues do students find helpful.* The research questions were answered with a three-part research. The description of the data collection and analysis methods as well as the discussion about the quality of the procedure are in chapter 6, section 6.1. The content of this chapter has been published in two conference papers (Kinnunen and Malmi 2006; Kinnunen and Malmi 2008).

7.1 Part I: Survey of students who dropped out

The first part of the research focused on the students who dropped out of the course in spring 2005. The aim of the study was to answer the question *why do students decide to drop out of the CS1 course*? Information about the reasons for dropping out was collected with a questionnaire. The aim was to learn more about why students decide to drop out, and to receive course feedback from the students who do not usually fill in the course feedback forms (because they are not in the course anymore at the time the course feedback form is published). However, the analysis focused primarily on aspects that shed light on the drop-out phenomenon and the feedback related answers were left to lesser attention.

In spring 2005, 560 students enrolled in the introductory programming course (CS1). To avoid ambiguity, the number of students is defined as the students who received some points from the exercises. The CS1 course at Helsinki University of Technology (TKK) is a large-scale course; each year some 500 – 600 students enter the course and 10% of them never submit any exercises during the course. Therefore, in this study, all statistics are collected for those students, who at least submitted something for the course.

In spring 2005, 37.9% (N = 212) students failed to get a passing grade from the course and thus dropped out of the course. The questionnaire was sent to those 212 students who had enrolled in the CS1 course at the beginning of spring 2005, but had failed to finish the programming exercises and/or the project work. The questionnaire was sent at the end of the course. The first reminder was sent a couple of weeks later and the second and the third reminders were sent at the beginning of the fall semester. Altogether 105 (49.5%) students answered the questionnaire.

The questionnaire included both closed-ended and open-ended questions concerning the course content, study material, teaching, perceived difficulties and reasons for dropping out. Open-ended questions were merely used as ”Other reason, please elaborate” -type of questions. The questions were formed based on previous course feedback, previous studies on aspects that may affect the students' decision to drop out, and anecdotal information concerning the difficulties students confront during the course.

7.1.1 Background information

Out of the 105 respondents, 75 (75.3%) were male. According to the Chi-square test for cross tabulations of the gender distribution, the female respondents were overrepresented in the data. The respondents' year of study ranged from first year to the fifth year or more (Table 15). However, the most of the respondents were studying in their second or third year at the university. The most common study programmes among respondents were Electronics and Electrical Engineering, Communications Engineering, and Mechanical Engineering. In total, respondents represented 15 different study programmes. The distribution of the three largest degree programs among the respondents was statistically significantly similar to the distribution of degree programs among all dropped out students at the course (χ^2 (2) = 5.00, $p < 0.05$). The reason for considering only the three largest degree programs when examining the similarity of the distribution was that the three largest degree programs covered 60% of the respondents' degree programs. The rest of the answers split into 12 degree programs thus resulting in low frequencies.

Table 15 The respondents' year of study

		f	%
Year of study	I	6	6,5
	II	38	40,9
	III	28	30,1
	IV	7	7,5
	V/V+	14	15,1

For the majority (69.5%, n = 66) of the respondents, the course was compulsory. 34.0% (n = 32) of the respondents had plans to take advanced programming courses while 63.2% (n = 60) of respondents stated that they had friends in the same course. Majority of the respondents (63.8%, n = 60) stated that they did not have any programming experience before this course. The ones who reported previous experience were asked whether they had earlier attempts to take CS1 course at Helsinki University of Technology (TKK) or whether they had taken some other programming course at other university or high school. (Table 16).

Table 16 The respondents' background information

	yes		no	
	f	%	f	%
Compulsory course	66	69.5	29	30.5
Plans advanced courses	32	34.0	62	66.0
Friends at the course	60	63.2	35	36.8
Programming experience	34	36.2	60	63.8
Previous attempts to take CS1	16	44.4	20	55.6
Other programming courses	16	43.2	21	56.8

Java and Visual Basic were the most commonly used programming languages among students who had some previous programming experience (Table 17). 73.6% (n = 25) of students who reported previous programming experience had made programmes that were less than 200 lines. 55.8% (n = 19) also reported that they were not familiar with object-oriented programming before enrolling the CS1. The rest of them had gained some familiarity from previous attempts to take the course or some other way. On average, the students' had a limited programming experience.

Table 17 Programming languages students, who had previous programming experience, had used prior to enrolling their first programming course

		f	%
Programming languages	Java	15	44.1
	Visual Basic	11	32.4
	Pascal	9	26.5
	C++	9	26.5
	Basic	8	23.5
	C	5	14.7
	Perl	1	2.9
	Python	0	0
	Delphi	0	0
	Other	6	17.6

7.1.2 Study skills

The questionnaire mapped out students' own estimation of their study skills and time management skills (Table 18). The respondents were given a list of options to choose from one or more statements that described them best. Therefore, the sum of responses can exceed the number of responses to the question (n = 90). In the table, f stands for the frequency. The table discloses that students chose statements that equate with inadequate study skills more often than statements that coincide with advanced study skills.

Table 18 Students' evaluation of their own study skills

Advanced study skills	f	%
I am usually good at estimating how much time courses take	40	44.4
I usually start doing exercises/preparing for exams well in time	19	21.1
I know how to adapt different study strategies according different courses	14	15.6
sum	73	

Inadequate study skills	f	%
I often underestimate how much time courses take	32	35.6
I usually start doing exercises/preparing for exams at the last moment	49	54.4
I would like to get more information and guidance about study strategies	19	21.1
sum	100	

Respondents were also asked whether they attended the lectures and exercise groups arranged by the course personnel, whether they studied the course slide, books, educational material provided by the teacher or other additional material, such as web pages, and did they use course news group as an information source. Each previously listed activity is regarded as available and authorised information source for the student.

The total number of the information sources the respondent reported therefore reflects the quantity of possible sources where help was available in case of students confronted difficulties. The mean value for the number of the information sources was 4.3 and the mode value was four. The most frequently used information sources were the slides and attending lectures. The least used information sources were reading the books or other educational material provided by the teacher (Table 19).

Table 19 Used information sources

	f
Slides	78
Lectures	75
News group	62
Exercise groups	61
Other (web)	49
Educational material	48
Book	37

Students were also asked how many study weeks[13] they were planning in January to get during the spring semester and how many they eventually got (Figure 34). The mean for planned study weeks was 18.3, mode value being 20. Correspondingly, the mean for actual study weeks students received was 10.1 and mode value was zero. The difference was found to be statistically significant ($Z = -4.80$, $p < .001$) even though the effect of CS1 course was eliminated. Therefore, the results suggest that students had more general type of difficulties with their studies than just CS1 related difficulties.

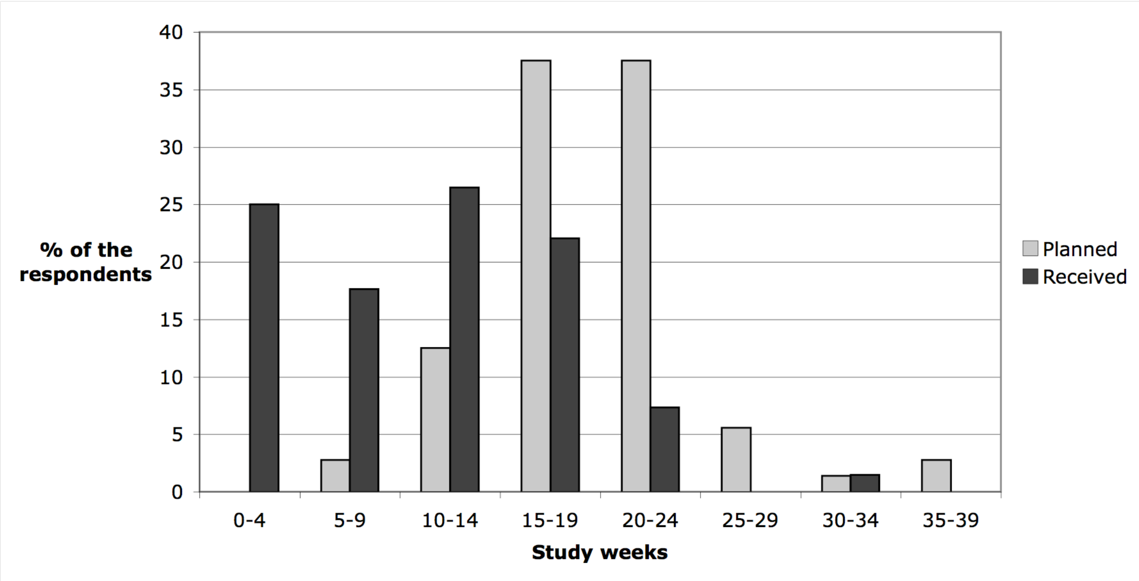


Figure 34 Planned and received study weeks of students who dropped out of the CS1 course in 2005

[13] A study week corresponded 40 hours of work.

7.1.3 The drop-out phase

Getting a passing grade from the course required passing the programming exercises, project work, and the exam. The statistics from the previous years revealed that the exam was not in the key position when considering the drop-out phenomenon. Therefore, the focus was turned on the programming exercises and the project work. 44.8% (n = 47) of the respondents dropped out of the course during the programming exercises, 7.6% (n = 8) finished their exercises but did not start the project work, and 35.2% (n = 37) dropped out during the project work. Therefore, roughly half of the respondents dropped out of the course during or right after the programming exercises and the other half dropped out during the programming project.

The Kruskal-Wallis test revealed that the fact that the course was mandatory, plans concerning advanced courses or previous programming experience did not have statistically significant affect on the phase when students dropped out of the course. However, students who had friends at the same course dropped out of the course later than peers without friends at the same course (Table 20). The difference was statistically significant (χ^2 (1), $p < .01$).

Table 20 Drop-out phase and friends

	Had friends		No friends	
	N	%	N	%
During programming exercises (N 49)	22	22,4	26	26,5
After programming exercises (N 47)	38	38,8	9	9,2

I was also interested in whether the number of the information sources that students used during the course and the reported number of the reasons for dropping out correlated with the drop-out phase. The number of the information sources was derived from several variables (lecture attendance, exercise group attendance, studying slides, studying educational material, reading the book, using some other information source, such as internet, and seeking help from the course's web based news group). For example, if the student attended at least some lectures and read the book the number of the information sources was two. The number of reasons for dropping out was derived from the number of the reasons the students chose in the questionnaire or wrote down in the open-ended question. The correlation matrix suggested that only the number of reasons for dropping out had a very weak linear connection to the drop-out phase ($r_s = -.33$, $p < .01$) (Table 21). Those students who dropped out at the early state of the course reported more reasons than students who dropped out later did.

Table 21 The correlation between the drop-out phase, planned study weeks, number of information sources and number of reasons for dropping out

	Drop-out phase	Planned study weeks	Number of information sources	Number of drop out reasons
Drop out phase	1.00			
Planned study weeks	.18	1.00		
Number of information sources	.27	.02	1.00	
Number of drop out reasons	**-.33**	-.03	-.08	1.00

7.1.4 Reasons for dropping out

From the given list of reasons there were three main reasons for unfinished programming exercises: *other courses demand time*, *lack of knowledge*, and *time management* issues (Table 22). Other reasons covered topics, such as work commitments (4 mentions), programming exercises were too difficult (3 mentions), assignments were obscure and schedule conflicts (one mention each). There were also six mentions concerning the long time students had to wait in the exercise group[14] to get help. Three students complained that the assistant teachers were not able to teach the course material.

Table 22 Reasons for not finishing the programming exercises

	f
It took too long time to do exercises and I dropped out to be able to concentrate on other courses	38
I did not know how to do programming exercises	26
I started to do programming exercises too late and thus I was not able to finish them by deadline	24
I did not understand the topics that were covered at the course	12
I was not able to use as much time for the course than I had planned	5
Personal/family reasons	5
Other reasons	14

From the given list of reasons why programming project was not finished, the most common reason was lack of knowledge and skills (28 mentions). The difficulties with time managing came as a second common reason (12 mentions) (Table 23).

Table 23 Reasons for not finishing the project work

	f
I did not know how to do the project work	28
I started to do project work too late and did not have time to finish it by the dead line	12
I did not get enough help when I was doing the project work	8
I estimated the project work would take less time to complete	6
Other reason	31

Finally, as a summary Table 24 presents the frequencies of the reasons for dropping out the course that students chose from the given list. Respondents were able to choose more than one reason from the list; the most mentioned reasons concerned the time. Students felt that the programming exercises and the project work took too much time and therefore they dropped out to be able to concentrate on other courses. The difficulties with understanding the course content and the actual programming were the second greatest reason to drop out. The respondents who chose the option "other reason" were encouraged to elaborate. The reasons that emerged from open-ended question concerned the difficulty and the size of the project work (10 mentions), lack of motivation (3 mentions), work commitments (3 mentions), getting sick or other

[14] Exercise groups were arranged several times per week. Students worked independently in a computer classroom. One to three assistant teachers were present to help students who needed assistance.

personal reason (2 mentions), group work exercise (2 mentions), and lack of help (1 mention). Examination of the answers on this questions revealed that many respondents chose more than one reason to drop out of the CS1. The number of the reasons for dropping out ranged from a total of one to five reasons. The mean of number of the reasons for dropping out was 1.8.

Table 24 Reasons for dropping out of the course

	f
It took too long time to do the project and I dropped out to be able to concentrate on to other courses	31
It took too long time to do the programming exercises and I dropped out to be able to concentrate on other courses	30
I did not know how to do programming exercises	21
I did not understand topics that were covered at the course	18
I was not able to use as much time for the course as I had planned	18
Personally/family reasons	10
I did not have time because I started to prepare for entrance examination[15]	4
I was not planning to continue my studies at TKK so it did not matter whether I got a grade or not	3
Other reason	24

7.1.5 Summary of the dropped out students' questionnaire

The students' self-evaluation concerning their own study skills emphasised that many of the respondents perceived their skills to be inadequate. The difference between the planned and received study weeks during the spring 2005 corroborates this perception.

Approximately half of the respondents dropped out of the course during the first half of the course right after the programming exercises. The students who had friends in the same course tended to drop out later than the students who did not have friends in the course. In addition, the students who dropped out later tended to have fewer reasons for dropping out compared to students who dropped out earlier in a course. However, whether the course was compulsory or not, previous programming experience, plans concerning taking the advanced programming courses, planned study weeks, and the number of informational resources utilised by the students did not relate to the drop-out phase.

The most reported reasons for not finishing the programming exercises and project work concerned time management issues, student's study preferences and inadequate knowledge and skills. The reasons that contributed to the final decision to drop out of the course related to the time, student's preferences, and inadequate knowledge and skills.

7.2 Part II: An interview of students who dropped out

The aim of this interview study was to answer the questions *why do students decide to drop out of the CS1 course* and *which content-related issues do students find difficult?*

[15] Some students had decided to apply for another university. In order to receive a student place they needed to take a part in the entrance examinations at the beginning of the summer.

For the questionnaire, sent to the students who dropped out of the CS1 course in spring 2005, there was also an inquiry concerning students' willingness to be interviewed about the course in general and the reason for dropping out of the course. Twenty-two students out of the 105 who responded expressed their willingness to be interviewed; eventually, 18 students were interviewed during the fall 2005 and early spring 2006. The remaining four students were not interviewed due to incompatible schedules or the students' decided they did not want to come to an interview after all. All interviews were held in Finnish, tape-recorded (with the exception of four cases when the interviewees did not consent to recording or technical difficulties). In addition, notes were taken during all interviews. Finally, tapes were transcribed by the author. A more detailed description of the data collection and analysis methods as well as the discussion about the quality of the procedure is in chapter 6, section 6.1.

7.2.1 Background information

Out of the 18 students interviewed for this study, ten of them were studying their second or third year and the rest of them had been studying four or more years. Six (33.3%) of the interviewees were female and 12 were male. According to the Chi-square test, the gender distribution of the interviewees was similar to the gender distribution of all the students who dropped out of the course (χ^2 (1) = 3.60, $p < 0.05$). The interviewees represented six different degree programs. The three most common degree programs were Electronics and Electrical engineering (seven interviewees), Mechanical Engineering (five interviewees), and Communications engineering (three interviewees).

For majority of the interviewees (n = 12) the CS1 course was compulsory and for the rest it was optional. However, most of the students were not required to take any other programming courses during their studies if they do not choose otherwise. 11 students reported that they had at least some previous experience on programming before entering the spring 2005 course. They had gained that experience either from earlier attempts to get a passing grade from the course, high school courses or from the courses at university of applied sciences. Six of the eleven reported that they were somewhat familiar with object-oriented programming before they entered the course. Nine students reported that they have plans to take advanced programming courses. At the beginning of the spring semester in 2005, students planned to study in average 21.7 study weeks. The range of planned study weeks was from 8 to 39.

7.2.2 Reasons for dropping out

The interview data suggest that there were two major reasons for dropping out: 1) *no time* (12 interviewees mentioned this) and 2) *no motivation* (5 interviewees). However, it is more interesting to understand why students do not have time and motivation. In addition to these two major categories, there were other smaller categories mentioned as well. Almost all interviewed dropped out students gave more than one reason, which indicates the complexity of the problem. Each reason to drop out will be discussed separately and only thereafter I will concentrate on real life cases where the accumulation of reasons can be seen.

Category 1: No time

This category could be divided further into three subcategories, which were: No time because of

1.1 The student decided to prefer doing something else

1.2 The student had not allocated enough time for the course in the first place

1.3 Some parts of the course were more difficult than what students expected and therefore the course took more time than planned

The first subcategory includes reasons such as:

- A student has many assignments, which conflicted with *other courses* (courses that are compulsory and are prerequisite to other courses and therefore it is more vital to pass them instead of CS1)
- It is possible to postpone the course without *any consequences* (for example, CS1 is not prerequisite to any other course, or course content is not likely to change much over the years)
- The *course is optional* (some students took the course out of curiosity or they though that programming skill might be useful in a future. All the same, since the course is optional it is first to go when the timetable gets booked up).
- A student has a *job/hobby related commitments* (for example, has to leave for a long work trip in the middle of the course)
- A student decides to use time to prepare for *entrance examination* of another university. The following texts are translations of the transcriptions of the interviews.

Student 4: I had really a lot of courses [during the spring 2005]. I had greedily taken little bit too many [courses], over 20 study weeks. And then there were so many other exercises and exams and then as this is an extra [optional] course for me. So, I took this course merely out of my own personal interest. As all the exercises started to accumulate, I had to drop something out. As this wasn't compulsory course, I decided that I rather drop out this course than risk all my compulsory courses.

The same interviewee continues:

On the other hand I had already made those programming exercises meaning that I already knew the basic stuff and I had already gained the goals I had set to myself concerning this course. If I think about my studies I don't actually need programming but if we have to make all sorts of programs for machine tools and others then it is good to understand all this, how to make subprograms and how to use them in main program and all these loops and others. To understand how they are constructed. That is useful to know and otherwise it has also helped me to understand. In that way, I had already attained my goals. I don't know how much added value the programming project would have brought. Well, of course I would have got the study weeks.

The second subcategory was *no time because one has not made enough time in the first place*. The next sample is from a student who had started the CS1 course several times but had always dropped out of it at an early phase of the course.

Student 12: ...in a way I never stopped to think how much time this course would take in a worst case and how much I should invest to it in order to pass the course. I always dropped out at the early stage of the course. That is why I never got the proper picture of how much time it could require. So, that is why I [dropped out] so many times.

The third subcategory, *some parts of the course are more difficult than expected* and therefore the course takes more time, I will look at closer a bit later.

Category 2: No motivation

The other major category *no motivation* was also divided into subcategories, which explain/give reasons why some students did not have enough motivation to finish CS1 course. Three subcategories emerged:

2.1 No study motivation in general,

2.2 Payoff is imbalanced and therefore motivation drops,

2.3 Some parts of the course were too difficult and therefore motivation drops

The first one of these included different cases such as the student was not at the department he/she wanted to be. For example, the student was studying at the Electrical and Communications Engineering department because he/she did not have high enough entry scores to study the area he/she really was interested in and therefore had more general difficulties to orient to university studies.

> *Student 14: Well, I didn't have any big courses at the same time or other courses in general either. ... my study motivation was low in general. That is why, that is probably why I decided to drop out of the course.*
>
> *Interviewer: Have you though of what might have caused that your study motivation dropped?*
>
> *Student 14: It was probably because of I was in the wrong department. Those studies there didn't interest me. That was probably mostly it.*

The second subcategory concerned the experienced imbalance between the amount of work needed and the gains achieved. Students felt that the course required more than other courses with nominally the same extent in study weeks. The source of imbalanced payoff concerned both the time and effort required and the achieved level of the programming skill.

> *Student 4: My motivation dropped because the workload would have been so huge compared to what I would have received [ECTS units]. Since I wasn't short of study weeks. My study grants were not hanging on these study weeks.*
>
> *Student 12: Well, sometimes it was like that I asked [the assistant teacher] many times and it [programming exercise] got just a little bit further. Then it took several hours of work and two, three visits at the programming exercise group that I was able to finish one programming exercise. In addition, I had been thinking about it by my self. I felt so frustrated that I spend almost ten hours to the program that counts the mean of three numbers. There is no creative thinking there. It just... fighting that you get the program to work.*
>
> *Student 3: 5 study weeks doesn't apply to me. It should be 10 study weeks if you use the time used to the course as a dipstick.*

The third subcategory, some parts of the course were too difficult, I will discuss later. These reasons are also related to category 1.3.

Category 3: Desire to receive a good grade

Two students decided to drop out of the course because it did not look likely that they would pass it or would pass it with a good enough grade. These students were ambitious and wanted to understand and pass the course with a good grade instead of merely just passing.

Category 4: Difficulties with connecting with other students

At the few latest programming exercise round there was a planning exercise that was to be made in a group of two to four students. It was possible to pass the exercise round also by doing all the other exercises on that round but that required very high scores on those remaining exercises. Some students found it difficult to find a group where to make the planning exercise. These students did not have any friends at the course. In some cases, the age difference between students increased the difficulty to find the group (a few students were over 20 years older than the others).

> *Student 1:.. I didn't like the group works that we were supposed to. It didn't cover actual programming I generally don't like group works I decided already at the beginning of the course that I would not do those exercises at all.*
>
> *Interviewer: Did you have any friends at the course?*
>
> *Student 1: No I didn't. That was the main reason why I didn't do those exercises that required group work. Because I didn't know anybody from there.*

The age difference also affected in the more general level. Two interviewees reported that they were discriminated by other students because they were more mature than the majority of students.

> *Student 3: I have noticed a lot of age racism here.*
>
> *Interviewer: Especially here at TKK?*
>
> *Student 3: Yes. During the first study years there were very few people who wanted to know me we sit day after day next to each other at the lecture hall. For example, if we arrive the lecture hall at the same time they look the other way. I can't even say hello because they don't give me the opportunity for eye contact. And I don't want to hurt anybody's feelings by shouting hello if they don't want it. The situation is different if they are tiddly or they need help. Then they know me from far.*

Another student has similar experiences.

> *Student 8: Students are not used to more mature students. Until now they have only been at school with coevals. The majority of students are males aged between 19 and 25. They think they know everything and they tend to rate every human relation as possible/not possible courtship relation. In addition to that, they are still in a phase where they are becoming independent. A female student who is almost the same age than their mother — well that is something they want to get rid of.*

These were not primary reasons for dropping out, yet they affected students' general comfort level and therefore, combined with other difficulties, had an influence on dropping out.

Category 5: Plagiarism

Besides these two major categories and their subcategories, there were other reasons for dropping out. In two cases students had to drop out after they had been caught plagiarising[16] their programming project. One student explained plagiarism with more liberal study culture at other departments saying that teachers are looking the other way and that collaboration is informally, if not formally, allowed. The other student explained that he and his friend did not know that plagiarism/group work was not allowed and that they did it because programming project was hard.

[16] Here plagiarism refers to unauthorized group work when doing a programming assignment or programming project. It also refers to copying the code of another student and representing the code as one's own work.

Category 6: Degree structure reform

One student dropped out because he was in a wrong course. This was due to misunderstanding concerning degree structure reform and its effect on students' studies. This change may have confused some students, since not all of them were aware of whether they needed this course or its parallel course in order to complete their studies.

7.2.3 What students consider difficult at the introductory programming course

Many students said that the course or a part of it was hard. This experienced difficulty was partially connected to both *no time* and *no motivation* as reasons for dropping out. Difficulty was in some cases the reason why the course required more time than anticipated. It also affected negatively on motivation. During the interviews, students were encouraged to expand on what aspects of the course they found particularly difficult. The following difficulties emerge:

1) Difficult to debug:

Interviewees mentioned that they found it difficult to find errors in their code. They got frustrated as the error was usually a minor/trivial one and it still took hours to find it.

> *Student 1: Yes, there was enough [time to complete exercise]. But then when you can't figure out some occasional point and you try to figure it out one day after another. That can result that you run out of time.*

> *Student 6: I wasn't able to figure out where the problem was. I was able to check the code until the object came. But then the automated assessment system didn't give me any points even though I though that the most of my code was correct.*

An example of the type of error that required a lot of time to discover:

> *Student 9: Some things were in a wrong order when doing those loops, like some for loops and similar. They often had some minor things, such as, how it goes though those elements. In those situations I had somehow defined it wrong so that it goes over or something similar. It was quite hard to think what happens in a loop. It might have been a small error and compiler didn't tell so exactly where the mistake was even though it told the location of the error. This kind of things took a lot of time, as you didn't get any help"*

2) Managing extensive unity:

For some students, the size of the project work was difficult to handle.

> *Student 12: It was the programming project, as you had to do a larger unit. At that point I stared to freeze. It was so hard to implement all those things I had learned earlier. "*

> *Interviewer: So, was it difficult because it was a large unit or because you had to write more complex code or what was the thing that made it difficult? "*

> *Student 12: Well, it was such a big unit altogether. It felt so extensive job to do. Even if you aim for the lowest grade you have to do a lot of work. At least I spend quite a lot of time doing it.*

3) Not enough knowledge/ability:

Some students felt that their knowledge on programming concepts and their programming skill were insufficient.

> *Student 3: It was so laborious to do the programming project considering the knowledge that had been given so far. There remained a lot of rustling up to do. I know how I could have managed to do the project but since it was not absolutely essential for me to do it....*

> *Interviewer: Was the laboriousness of the project the crucial thing here?*

Student 3: No, not the laboriousness but the lack of knowledge. My subject was a family tree and it proved to be pretty hard to do those branches...

4) Assistant teacher was not able to help:

In some cases students felt they did not get enough help from the assistant teachers.

Student 2: I didn't get what I was looking for at the programming exercise groups. I think it is not enough that tutor knows something. He/she has also been able to mediate that knowledge to the student. [The assistant teacher] has to be able to analyse student's questions and based on that insight and assistant teacher's own knowledge about the subject matter be able to tell the answer to the student so that he/she understands it. That is something I did not notice there. There were only two assistant teachers who were something of the sort of teachers.

Interviewer: So we should invest some energy in assistant teachers' pedagogical education?

Student 3: Yes, definitely. They [assistant teachers] should not just snatch the keyboard and start to potter. And finished!

Student 3: Many [assistant teachers] just were there. They were able to cope with the keyboard but things didn't get any clearer for me. I was left alone to wonder about the code There was an incident when assistant teacher came and messed up the code and then left because he /she couldn't figure it out.

7.2.4 The diversity of the reasons for dropping out

As noted already earlier, many students have reported several concurrent and cumulative reasons for dropping out. In addition, there are issues that affect the final decision to drop out indirectly. For example, there might be several small issues that would not alone cause the student to drop out of the course, but together the reasons cumulate and their importance in their final decision to drop the course grew. To shed some light on the multidimensional nature of the students' reality, I used grounded theory (Strauss and Corbin 1990) and an ad hoc approach to analyse the interview data (Kvale 1996). The transcripts were read together with the answers students had given to the questionnaire. I was constantly trying to gather coherent stories about what lead each student to drop out. As a result, each interview was summarised into a table where each column included information concerning student's previous programming experience, his/hers study related plans, motivation, social aspects, studying habits, timetable, course related issues and actual studying. The data was arranged so that the first row described the time before the course and following rows equalled different parts of the course, such as programming exercises and project work, see an example in Table 25. This manner of representation enabled to detect coexistence issues that affected student's studying process. All tables, except for the one (Table 25), are in Appendix 8.

Subsequently, the tables were analysed using the model of the instructional process as a guiding tool. The aim was to highlight how the reasons for dropping out place themselves to the model of the instructional process. For example, student 18 stated that the following facts affected his/hers decision to drop out: 1) he/she had other courses at the same time, 2) he/she got a low points from the project work's planning phase, 3) he/she felt that the grading was unfair, and 4) he/she got irritated by non-constructive comments given by the assistant teacher. In addition, he/she felt that his/hers skills were not good enough to cope with the project work and he/she had an opportunity and time take the course later.

Other method that I used to analyse and visualise the cumulative nature of the reasons for dropping out was drawing the figures or diagrams of the aspects that had resulted student to drop out of the course. Each student's reasons for dropping out the course were presented as a visual diagram that showed the connection between different events. These diagrams also highlighted how the reasons for dropping out cumulated individually. There were no two similar figures in the data. Below is an example of two drop-out diagrams that were constructed during the analysis of the data (Figure 35). For instance, the lower diagram highlights that the student had three immediate reasons for dropping out the course. He had no time, he had the possibility to take the course later, and he wanted to use his time for other activities. A closer look at the reasons revealed that the reason for not having enough time was initially related to misjudgement of one's own abilities. The drop-out diagrams are not attached as an appendix since the same reasons for dropping out can be found in the tables (Appendix 8).

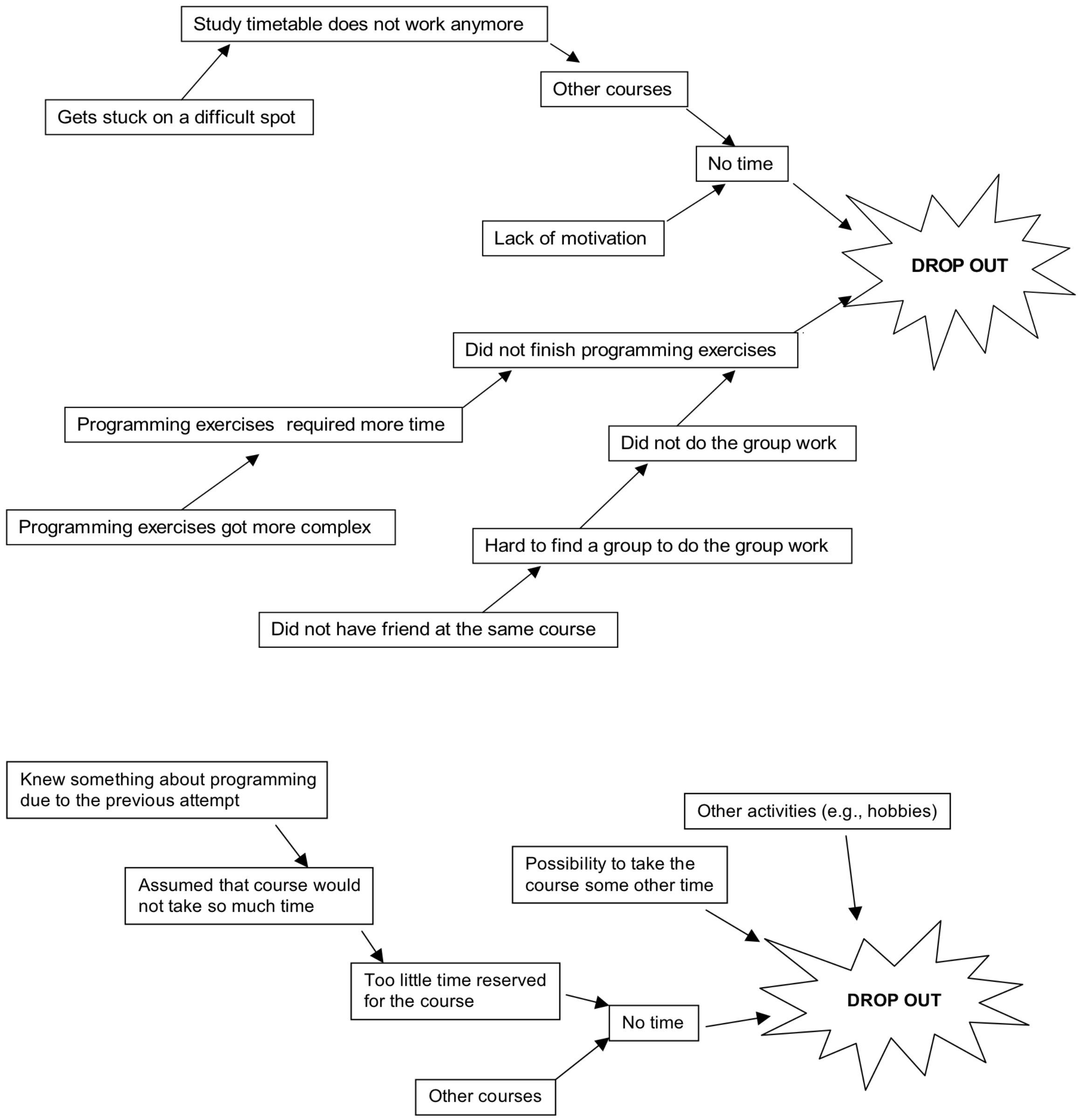


Figure 35 Cumulative causes of drop out

Table 25 An example of a table that summarises student's experience in the class (Student 18, the year of study II)

	Prior experience	Plans	Motivation	Social aspects	Studying habits	Course	Timetable	Studying
	Has limited programming experience from high school courses. Was familiar with Object-oriented programming.	This is a compulsory course. Plans to take advanced programming course. Plans to get 38 study weeks during the spring semester.		Had friends at the course.	Usually tends to start doing exercises in the nick of time. Usually underestimates how much time courses take. Is able to adapt different study strategies according to different courses.			
Programming exercises			Doesn't think that he/she is very good at programming, but is still interested in programming.	Feels that it is unnatural that you are not allowed to show your code as an example to others even when only helping a friend to get over a difficult spot.	Attends 0-20% of lectures. Attended some lectures but didn't feel like that was learning anything. Preferred studying the equivalent time independently. Studies slides, reads educational material, and uses other material such as internet pages. Didn't attend exercise groups because had friends who were able to help if needed. Seeks help from the news group.			
Project work		Had some free time in a following summer since didn't have summer job for the whole summer break. Plans to take the programming course in the summer.		Gets irritated by assistant teacher's non-constructive comments concerning the project work.		Gets a low grade of project plan and concludes that there is still a lot of work to do. Feels that the grading of the project plan was unfair.	Other exams and projects at the same time.	The course didn't give good enough skills and abilities to cope with the project work. Would have needed more help with the project.
		Receives 22 study weeks during the spring semester.				**Doesn't finish the project work and drops out.**		

7.2.5 Summary of the interview study

The aspects that influence students' decision to drop out are manifold and they tend to cumulate over the progression of the course. On the one hand, there were several concurrent reasons affecting student's decision (e.g., work and family commitments, difficulty of the course, and other courses that require time for studying). On the other hand, some reasons also cumulated creating a chain of reasons that affect student's decision to drop out of the CS1 (e.g., student had previous programming experience → student misjudged the time need for the course and reserved too little time for studying → student runs out of time and has to drop out).

The main reasons for dropping out are related to the lack of time and motivation. However, there are varieties of possible contributing factors that stand behind those reasons. The underlying reasons behind lack of time concern the student's preferences (e.g., preferences concerning other courses, work commitments, hobby), time managing difficulties, and the unexpected level of the difficulty of the course. The underlying reasons behind the lack of motivation concerned the low study motivation in general, perceived imbalance of payoff, and the perceived difficulty of the course.

7.3 Part III: Survey of dropped out and passed students

The third part of the study focused both on the students who dropped out of the course and on the students who passed the course. The questionnaire was build based on the results from the previous parts of the study. The aim of the study was to answer the following questions: *Why do students decide to drop out of the CS1 course? Which content-related issues do students find difficult? Are there statistically significant differences in what programming related issues students who drop out and students who pass the course find difficult? What kind of strategies to overcome difficult issues do students find helpful?* A more detailed description of the data collection and analysis methods as well as the discussion about the quality of the procedure is in chapter 6, section 6.1.

In spring 2006, 564 students started the CS1 course. 26.1% (n = 147) of them did not get a grade from the exercises and thus dropped out of the course. At the end of the course, an email with a request and a link to the questionnaires was sent to both students who passed the course and those who dropped out of the course. Three reminders were sent to those that did not respond. 47.5% (n = 198) of students who passed the course answered the questionnaire. At the same time, 25.9% (n = 38) of the drop-outs answered the questionnaire.

The next year, 674 students started the course and 26.4% (n = 178) of them dropped out during the programming exercises. This year some alterations were made to the timetable when the questionnaires were sent. The drop-outs' questionnaire was sent for the first time already in the middle of the course to catch those students who I knew based on their performance they had no hope of passing the course. This alteration concerning the sending timetable was made to enhance the drop-outs' response rate. The other questionnaire for the students who had passed the exercises got their questionnaire at the end of the course as previous year. In 2007, 52.6% (n = 261) of the students who passed and 45.5% (n = 81) of the students who dropped out answered the questionnaire. In summary, the final data consisted of 459 answers from the students who had passed the programming exercises and 119 answers from the students who dropped out.

7.3.1 Background information

In 2006, 72.1% (n = 148) of the respondents in the passed students' questionnaire were males compared with 73.9% (n = 193) in 2007. The majority of respondents for the drop-outs' questionnaire were also males; 84.2% (n = 32) in 2006 and 75.9% (n = 60) in 2007 of respondents were males. The gender distribution of the questionnaires is in Table 26.

Table 26 Respondents' gender distribution

	2006		Questionnaire for			
	All students		Passed	Dropped out		
	f	%	f	%	f	%
Male	468	83.1	148	72.1	32	84.2
Female	95	16.9	49	24.9	6	15.8
	563		197		38	

	2007		Questionnaire for			
	All students		Passed	Dropped out		
	f	%	f	%	f	%
Male	533	80.5	193	73.9	60	75.9
Female	129	19.5	68	26.1	19	24.1
	662		261		79	

In 2006, the majority of the respondents' from both groups (dropped out and passed students) were studying their first or second year. In 2007, there were very few first year students among the respondents. Instead, in 2007, 33.6% of the respondents were second year students and 37.5% were fourth year students. Meanwhile, 20% were studying in their fifth year, or later, at the university. In 2006, the dropped out students got at least some points from, on average, 33.4% of the exercise rounds. In 2007, the corresponding percentage was 45.5%.

The respondents who passed the course represented 19 different degree programs. Correspondingly, the students who answered the dropped out students' questionnaire represented 15 different degree programs. In both cases, Electrical Engineering was the largest study program. Machine Engineering, Communications Engineering, and Civil and Environmental Engineering followed next in second and third place.

Pearson's Chi-square test was used to examine whether the respondents' gender, year of study and degree program distributions were statistically significantly similar or different from the corresponding distributions of the all passed/drop-outs in the course; the values of Chi-square are summed in Table 27. The respondents' gender distribution reflected, somewhat the overall gender distribution of the CS1 course. However, in 2006, females were overrepresented in the category of the students who passed. In 2007, females were overrepresented in the category of the drop-outs. The distributions of the year of study among respondents and all passed/dropped out students at the course were similar and no statistically significant differences were found.

In most cases, the respondents' degree distribution was similar to the distribution of degree programs of all passed/dropped out students at the course. Only the three largest degree programs were taken into account for this test since the students typically tended to scatter over several degree programs and thus resulting low frequencies. The three largest degree programs also covered over 50% of all respondents. The results of Pearson's Chi-square test suggested that the most distributions of degree programs were well in line with the distribution of degree programs among all passed/dropped out students in the course. One exception was the questionnaire that was sent to the students who passed in 2007. In that year, the Civil and Environmental Engineering was slightly overrepresented.

Table 27 Results of Pearson's χ^2 tests

	2006 Dropped out	2006 Passed	2007 Dropped out	2007 Passed
Gender	χ^2 (1) = 0.00***	χ^2 (1) = 8.66*	χ^2 (1) = 9.27*	χ^2 (1) = 2.72*
Year of study	χ^2 (2) = 5.85*	χ^2 (4) = 2.04*	χ^2 (2) = 2.44*	χ^2 (4) = 4.72*
Degree program	χ^2 (2) = 1.67*	χ^2 (2) = 1.38*	χ^2 (2) = 3.22*	χ^2 (2) = 12.50*

Note. *p < 0.05. *** p < 0.001

The model timetable for studies suggests that students should study approximately 30 ECTS credits per term in order to graduate in the planned time (5 years for a master's degree). However, internal reports of the university disclosed that very few students were actually able to keep up with the model timetable[17] (Rantanen and Liski 2009). The data in the study revealed that students who dropped out of the course planned, on average, to study less in terms of credits than students who passed the course. The difference between the number of planned and received credits among the dropped out students was statistically significant ($Z = -4.66$, $p < 0.001$) even though the effect of the target course on study points was eliminated. The mean for dropped out students' study weeks was 26.6. The mean for study weeks dropped out students' received was 18.2. However, the same was true for the passed students' planned and received study points, as well ($Z = -5.70$, $p < 0.001$). Passed students' planned study weeks' mean was 28.9 and the received study weeks' mean was 25.1. The distribution of planned and received study weeks are in Figure 36 and Figure 37.

63.9% (n = 221) of the passed students and 63.4% (n = 52) of the dropped out students reported in an open question that they enjoyed and felt at home at this university. There was a slightly greater amount of students who did not feel at home among the drop-outs than the passed students, however according to the Mann-Whitney test the difference was not statistically significant. There was a very weak linear correlation between certainty of choice of study field and general study motivation ($r_s = -0.34$, $p < 0.01$) and between certainty of choice of study field and feeling at home at this university ($r_s = -0.36$, $p < 0.01$). Not surprisingly, the students who were sure about their choice of study field also tended to report better general study motivation. In addition, there was a connection between the students' thriving at the university and the certainty of the choice of study field.

The respondents were asked to estimate their own study skills by asking whether they think they usually start solving the exercises well in time and whether they are able to

[17] One obvious reason is that the model timetable is overloaded requiring an average 48,5 study hours per week.

estimate how much working time courses usually take. The majority of the passed (53.1%, n = 243) and the dropped out students (53 %, n = 62) admitted that they tend to start solving the exercises too late. On the other hand, 61.1% (n = 280) of the passed students and 57.8% (n = 67) of the dropped out students reported that they were good at estimating how much time courses usually take. According to the Mann-Whitney test the difference between the dropped out students and passed students was not statistically significant.

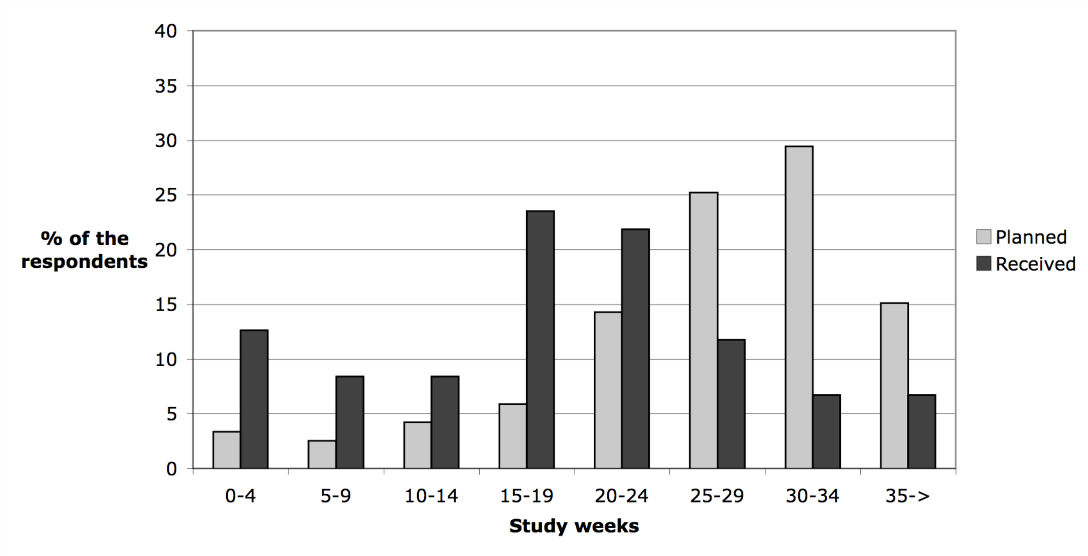


Figure 36 Planned and received study weeks of students who dropped out of the CS1 course in 2006 and 2007

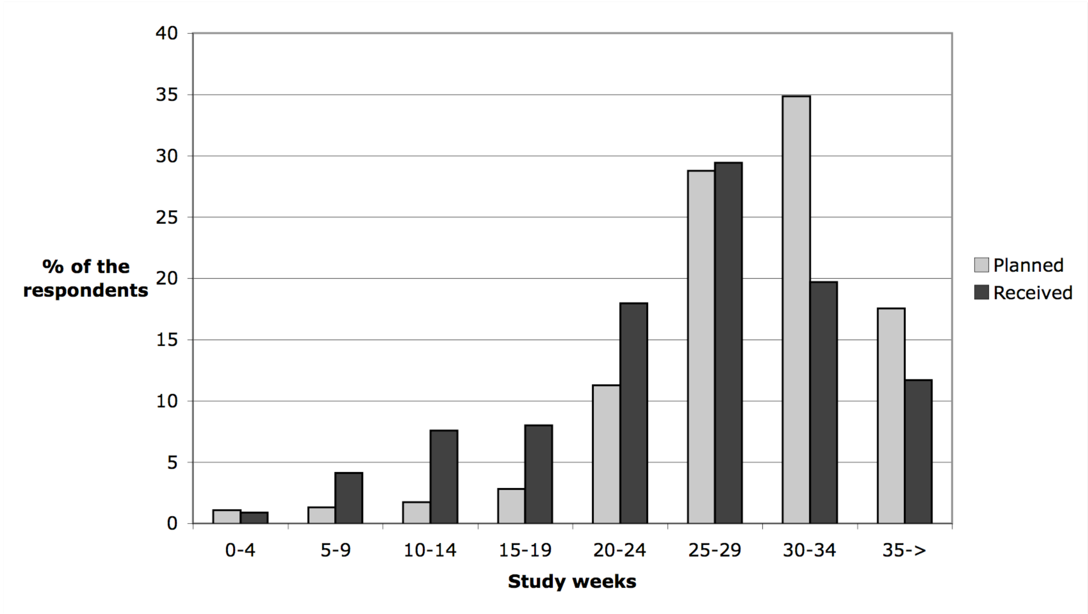


Figure 37 Planned and received study weeks of students who passed the CS1 in 2006 and 2007

7.3.2 Difficulty of programming related issues & strategies to cope with them

The respondents estimated the difficulty of the programming related issues by rating each listed item on a four-step fully anchored rating scale. In addition, the respondents were able to choose the option "I do not know". In that case, the response was regarded as a missing value in the analysis. The results are shown in Table 28. The higher the mean value is, the more difficult the issue at hand was perceived as by the respondents. The dropped out students experienced all aspects as more difficult than the students who passed the exercises did. The Mann-Whitney U test indicated that the differences were statistically significant in all but one case (using the text editor).

Table 28 Respondents' estimation on how difficult programming related issues were

	Passed	Dropped out	U	
Using text editor	1.4	1.6	17181.00	
Mathematics that is needed to solve the problems	1.4	1.7	18569.00	***
Testing own code	1.6	2.1	18064.00	***
Discovering a principle solution to the problem	1.9	2.2	21432.00	*
Adopting the exactness needed in writing program	1.9	2.4	13347.00	***
Understanding how given code is executed	2	2.6	16709.00	***
Adopting the programming style required at the course	1.8	2.6	13347.00	***
Identifying structures in a given code	1.9	2.7	15806.50	***
Discovering algorithm that executes the principle solution	2.3	2.8	16960.00	***
Finding compile time errors	2.5	2.9	17824.00	***
Transferring own thinking into programming language	2.4	3.1	15782.00	***
Designing parts of own code	2.4	3.1	15528.50	***
Designing the functioning of own code	2.3	3.3	14603.00	***
Finding run time errors	2.6	3.3	17684.50	***
ional statements (e.g., if-else)	1.3	1.7	17673.50	
Loops (e.g., while)	1.7	2.3	15085.50	***
Methods	1.8	2.5	14664.50	***
Ideas of OO	2.3	2.6	20058.50	***
Table	2	2.9	10076.50	*
Exceptions	2.2	3.2	5807.00	***
Inheritance and abstract class	2.5	3.3	10118.00	***
Handling files	2.4	3.3	8983.50	***

* $p < 0.05$, *** $p < 0.001$

The respondents were asked to elaborate in an open-ended question on what helped them get over the previously listed difficult issues. 62.8% (n = 243) of the students who got a grade from the exercises answered this question, whereas the response rate of the dropped out students was 37.0% (n = 44). I used a data-driven approach to analyse the open question answers. The procedure was close to what Ryan and Bernard call free list (2003). The written answers were categorised into 19 categories that emerged from the data. The procedure started by reading through the answers several times. The text quotes that expressed a strategy to get over difficult issues were placed into preliminary piles from which the categories emerged, as the process was iterated a couple of times.

Next, I looked for the common factors among the categories and ended up with six strategies (or meta-categories) that the students used to get over a difficult issue (see Table 29) The value f in the table refers to the frequency of how often the strategy at hand was mentioned.

Table 29 Strategies to get over difficult issues

		Passed		Dropped out		
Category		f	%	f	%	
Ask for help	Assistant teachers	62	27	6	14	
	Friends	45	20	10	23	
	Total	107	**47**	16	**36**	***
Study & think	Educational material provided by the teacher	42	18	6	14	
	Internet	23	10	2	5	
	Lectures and slides	21	9	6	14	
	Literature	11	5	1	2	
	Code examples found from different sources	8	3	1	2	
	Course news group	7	3	1	2	
	Studying, seeking answers (general)	24	10	12	27	
	Thinking, trying to understand	14	6	0	0	
	Total	150	**66**	29	**66**	
Practice	Doing exercises, practising	31	14	5	11	
	Trial and error	7	3	3	7	
	Total	38	**17**	8	**18**	
Persistence	Persistence & perseverance	38	17	3	7	
	Compulsory course	5	2	1	2	
	Total	43	**19**	4	**9**	
Manage time	Allocating more time for studying	18	8	4	9	
	Talking a pause	1	0	0	0	
	Total	19	**8**	4	**9**	
Interest & experience	Interest towards programming	4	2	0	0	
	Previous programming experience	5	2	0	0	
	Total	9	**4**	0	**0**	
Nothing	Nothing helped	2	**1**	9	**20**	

*** $p < .001$

Asking for help either from the assistant teacher or a friend was a much-used strategy among all students.

> *"... My friends were a great help. I did not think that the lectures helped me a lot. Together, we tried to understand things and we helped each other. If someone understood something, he then helped others."*

The next category is *studying* hard using different information sources (like course material, internet, and course newsgroup) and thinking about the difficult issue with the intention of really understanding. Many respondents stressed that they read the material over and over again until they understood. The following respondent uses his/her old programming exercise as a learning material.

> *"... sometimes I went through my programming exercises after I had got maximum points from them. I thought over what I had actually done, what happens where and why."*

The third strategy is *practicing programming* by doing exercises.

> *"The greater number of the programming exercises I did, the easier the problems I faced at the beginning of the course seemed to become. Difficulties I had at the last part of the course did not seem to be insuperable anymore like the difficulties I faced at the middle of the course. I started to internalise entities and discern Java's structure. The amount of practice was essential for getting over difficulties."*

> *"Great programming exercises! For example, a constructor-method thing got cleared up when I did the program that mixed drinks. The instructions for the exercise clearly implied that you use a constructor to create an empty class and you use methods to fill it with different drinks and to drink it."*

A subcategory of this third category is "getting over a difficult issue by doing exercises applying a trial and error approach".

The fourth category is *persistence*. The respondents reported that they had stamina and determination to work with the difficulties. They were self-disciplined and worked hard because they did not want to quit or the course was compulsory and they did not want to enrol in the course once again.

> *"Trying. I had taken a stand that I will do all the assignments. I forced myself to do them [exercises] and banged my head against a brick wall and I managed to do the exercises."*

The fifth category deals with *time*. Allocating more time for studying even at the expense of other courses helped the respondents understand difficult issues. One respondent also reported that taking a pause (like a good night's sleep) was helpful in some cases.

The last strategy deals with the personal *interest in programming and previous experience in programming*. In addition, especially the dropped out students reported that there was nothing that helped them when they faced difficulties and therefore they dropped out. The dropped out students' low response rate (37%) to this question could also be interpreted as a statement (if the student thought there was nothing that helped, he/she may have left the question unanswered).

The results suggest that the students who passed the programming exercises mentioned getting help, being persistent, and holding interest in the subject as viable strategies to get over difficult issues. However, the differences between the frequencies of mentions are in most cases not statistically significant or the significance was not calculated due to the low number of frequencies. The only exception is the "*Asking for help*" -strategy, which the students who passed mentioned statistically significantly more frequently to be helpful than the dropped out students (χ^2 (2) = 3.0E2, $p < 0.001$). On average, both dropped out and passed students mentioned 1.6 viable strategies to get over a difficult issue during the course.

7.3.3 Reasons for dropping out

The students who dropped out of the course were given a list of the possible reasons for dropping out and they were asked to rate how much each reason affected their decision to drop out. The four-step rating scale was fully anchored. In addition, the respondents were able to choose the option “I do not know”. In those cases, the response was recoded so that it equalised the neutral response. In Table 30 the low values represent low significance to the decision to drop out (1 = no effect at all) whereas high values represent critical reasons in the decision to drop out (4 = critical effect). The respondents rated the course workload-related issues as the ones affecting their decision to drop out the most. On the other hand, in general, being caught for plagiarism or plans to continue studies at a different university played a minor role in the decision to drop out.

Table 30 Reasons for dropping out

	mean
Doing exercises took too much time	3.4
Course required more time than other courses at the same study week quantity	3.2
Course's workload is not in balance with the payback	2.9
I did not know how to do programming exercises	2.8
I had reserved too little time for the 5 study week course	2.6
I did not understand the content that was covered at the course	2.3
I did not get enough help	2.3
Low motivation to study in general	2.2
Programming does not interest me	2.2
I wanted to concentrate on other courses	2.2
Dropping out does not affect my other courses	1.9
I had a lot of other courses at the same time	1.9
Course personnel's actions	1.7
Course arrangements	1.7
Personal reasons	1.7
Work related commitments took time from the course	1.6
Hobby related commitments took time from the course	1.6
This course is not obligatory for me	1.4
I am not going to stay at TKK	1.3
I had no intention to pass the course in the first place	1.2
I decided to start preparing for entrance exams	1.2
I got caught from plagiarism	1.2

The interviews during the previous parts of the research project revealed that students tended to have several concurrent reasons for dropping out; the quantitative survey corroborates that argument. On average, the respondents reported 10 reasons that had contributed to their decision to drop out at some level. Moreover, the respondents reported, on average, four reasons for dropping out that affected their decision critically.

In order to get a clearer picture of the drop-out phenomenon I decided to use factor analysis to reduce the variables. The procedure and pre-tests are explained in chapter 6.1. A Generalized least squares factor analysis followed by Varimax rotation reduced the list of drop-out variables to five factors, which explained 46% of the total variance. The first factor explained 10.5 % of the variance, the second 10.1%, the third 9.3 %, the fourth 9.0%, and the fifth 7.1% of the variance. The chosen model fit the data well (χ^2 (115) = 165.62, $p < 0.001$).

The variables that loaded high on the first factor dealt with course personnel’s actions and students’ perception of not getting enough help. Therefore, F1 was named *Course* *arrangements & not help*. The second factor had high loadings from the variables that considered difficulties understanding the course content. Therefore, the second factor was named *Difficulties in understanding course topics*. The third factor discusses the *Time management* issues and students' *preferences* for using time. The fourth factor had high loadings from variables like "The course is voluntary for me", and "Dropping out of the course does not affect other courses". Thus, the factor was named *Dropping out does not have consequences*. The fifth factor was named *Prefers other courses*. All factors and variables with the loadings are summarised in Table 31.

Table 31 Factor analysis' results of reasons for dropping out

Factors
F1: Course arrangements & not help (Cronbach's α = 0.83)
• The course personnel's actions (.86)
• Did not get enough help (.76)
• Course arrangements (.74)
F2: Difficulties in understanding course topics (Cronbach's α = 0.75)
• Did not understand subjects that were covered in the course (.99)
• Did not know how to do programming exercises (.74)
• Programming is uninteresting (.41)
F3: Time management and preferences (Cronbach's α = 0.40)
• Programming exercises took too much time (.71)
• The course required more time than anticipated (.64)
• I had reserved too little time for the course (.40)
• Personal/family reasons (-.36)
• Work related commitments (.30)
• Course workload is not balanced with the payback (.34)
F4: Dropping out does not have consequences (Cronbach's = 0.65)
• The course is voluntary for me (.74)
• Dropping out of the course does not affect other courses (.63)
• There was no intention to pass the course in the first place (.60)
• I am not going to continue studies at TKK (.50)
• I decided to use the time for preparing for entrance exams to continue studying elsewhere (.42)
F5: Prefers other courses (Cronbach's α = 0.71)
• I had a lot of other courses at the same time (1.0)
• I wanted to concentrate on other course (.57)

7.3.4 Summary of the passed and dropped out students' questionnaire

Students who dropped out the CS1 course perceived all course content related issues more difficult than the students who passed the course. Respondents reported six viable strategies to get over difficult issues that they faced during the course. The strategies were asking for help, studying and thinking, practicing, being persistence, managing time, and having an interest in programming. The students who dropped out found asking for help a less viable strategy than the students who passed the course.

The students who dropped out reported on average ten reasons that contributed to their decision to drop out. Moreover, they reported on average four reasons that critically affected their decision. Factor analysis revealed five reasons for dropping out. First, there were reasons that related to the course arrangements and to students' perception that they did not get enough help. Second, students had difficulties in understanding course topics. Third, students' time management skills and preferences played a part in the decision to drop out. Fourth, the fact that dropping out of the course did not have any consequences for the students made it easier to drop out. Finally, the students' preference for other courses affected some students' decision.

7.4 Summary of the difficulties students encounter during the introductory programming course

The three-part research revealed several aspects that affect the non-major CS students studying process. First, the introductory programming course is perceived as a demanding course. It takes time to finish all required exercises and the subject itself may be challenging to learn. Second, students' study skills and time managing skills are not always adequate to handle the many courses in student's study plan. Third, there are often several reasons for dropping out that contribute to the student's decision. Moreover, the reasons tend to cumulate in a unique way. This means that, on the one hand, students may have several reasons that are not related with each other affecting the decision at the same time. On the other hand, students may also have some reasons that cause other reasons, i.e. concatenations of events lead students to drop out.

The challenges that the students encounter during the CS1 course are manifold and they relate to the different phases of the instructional process as well as more general aspects that affect the whole process. For example, students may have difficulties at the planning phase when they are making timetables to handle all the courses that they are planning to take. The overall number of the courses may be too high, resulting inevitably in the lack of time at some point. Figure 38 visualises some of the aspects that play a role in students' studying process.

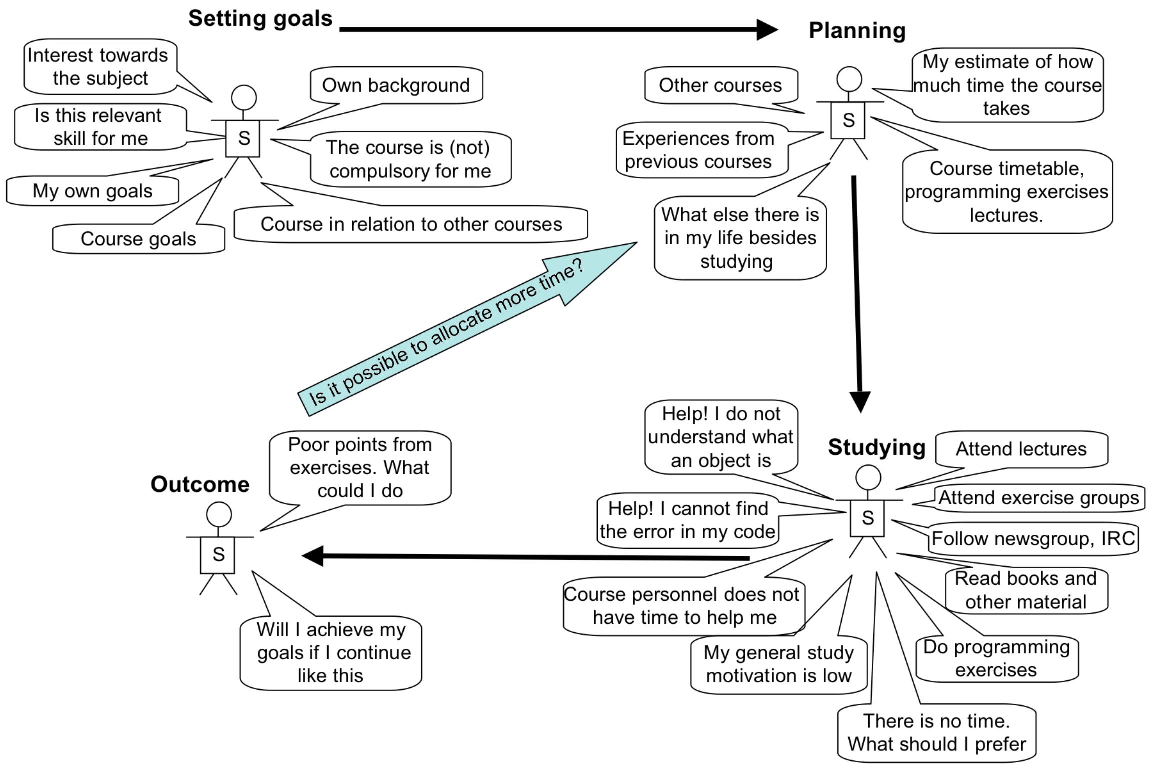


Figure 38 The studying process from the student's point of view

8 The instructional process from computer science teachers' point of view

The previous chapter discussed the studying process from the students' point of view thus answering the first research question. This chapter aims at answering the second research question: *How do CS teachers see the instructional process?* More precisely, this chapter focuses on teachers' conceptions of a variety of studying related aspects: *How do computer science teachers define what studying is? Which aspects of studying do computer science teachers think students have difficulties with?, Which aspects of studying do computer science teachers think are focal for successful studying? How do computer science teachers think that they can affect the students' studying process?* Thereafter, the focus is turned on the teachers' own teaching process and how they perceive it: *How do computer science teachers consider the different phases of the instructional process?* The data collection and the analysis procedures along with the quality discussion that relates to these results are described in chapter 6, section 6.2.

8.1 The students' instructional process from computer science teachers' point of view

The first part of this study focused at describing computer science teachers' conceptions of studying. The data was collected with a questionnaire that contained several open ended questions. The goals and the background of the study were introduced at a national small-scale seminar that was targeted for teachers and researchers interested in computing education. After the seminar, all 41 participants of the seminar were approached with an email that contained a link to the questionnaire. 25 teachers/assistant teachers out of 41 participants of the computing education seminar from 13 universities around Finland answered the questionnaire (Appendix 5). Thus, the response rate was 61.0%. 84% (n = 21) of the respondents were male. 16% (n = 4) of the respondents were Master of Science students, 44% (n = 11) had a Master's degree, one had a Licentiate's degree and 36% (n = 9) were PhDs. The vast majority (88%, n = 22) of the respondents had computer science background. Two respondents had studied other sciences and one respondent had studied mathematics.

16% (n = 4) of the respondents were working as research assistants, one was a researcher, 12% (n = 3) were teaching researchers, 48% (n=12) were lecturers, 8% (n = 2) were professors. None of the respondents was a planning officer or student counsellor. 12% (n = 3) had some other working title that previously mentioned. All the respondents had at least some first-hand experience of teaching. 60% (n = 15) of the respondents had worked as part time assistant teachers (a few hours per week), 80% (n = 20) had worked as assistant teachers (half or full time), 44% (n = 11) had been tutors, and 88% (n = 22) had worked as lecturers. The majority (88%, n = 22) had experience of more than one type of teaching relating positions. 36% (n = 9) of the respondents had taken some larger set of pedagogical courses, 40% (n = 10) had taken single pedagogical courses or studied pedagogical material unassisted. 24% (n = 6) did not have any pedagogical training.

8.1.1 Teachers' conceptions of what studying is

All respondents answered the question that asked the teachers to *define what studying is*. The responses to this open question were read through several times. After a while, differences of the focus within the answers started to emerge. Different focuses were arranged as preliminary categories during the first reading process. The reading and categorisation was repeated few times and original categories were rearranged until a simple and logical the model was created. Among the responses emerged three main categories that focused on A. knowledge & skills, B. action and C. student (Table 32). Each of the main categories was further divided into subcategories that emphasise different level of sophistication. The numbers inside brackets stand for the number of the quotations that fell into each category.

Table 32 Computer science teachers' definitions on what studying is

A. Knowledge & skill (7)	B. Action (13)	C. Student (7)
A.1 Building up knowledge (1)	B.1 Hard work (2)	C.1 A phase in student's life (2)
A.2 Acquiring knowledge and skills (4)	B.2 Collecting knowledge (2)	C.2 Student's skill (2)
A.3 Goal-oriented acquiring of knowledge and skills of entire discipline (2)	B.3 Goal-oriented actions (9)	C.3 Goal-oriented (mental) process that aims at understanding, internalising, applying (3)

The definitions in the category A emphasise the knowledge and skills. Quotations that fall into this category may also discuss the goals and some actions of how studying takes place, but the main focus is clearly on the knowledge and skills. The subcategory A.1 Building up knowledge expresses a straightforward view. Studying is something you do to assemble knowledge.

> *"Gradual assembling of the knowledge: first simple matters and then a little bit more complicated matters and so on…"*

The subcategory A.2 adds to the category A.1 by stressing that studying is about acquiring of knowledge and skills. The focus is on actively acquiring and digesting the knowledge and skills rather than simply piling up knowledge.

> *"Enhance know-how: understanding, internalising, applying facts and learning by using different learning methods."*

The last subcategory highlights the goal-oriented nature of the acquiring of the knowledge and skill. It is otherwise the same as the preceding subcategory, but it adds the goal-oriented nature of knowledge and skill acquiring.

> *"Absorbing the chosen field (or several fields) purposefully, both in the theory and in practice"*

The second category highlights students' actions. The first subcategory B.1 externalises simple view: studying is plain hard work. The second subcategory emphasises the collection of knowledge.

> *"Collecting explicit knowledge (active and passive), integration and acquiring it to one's own implicit knowledge."*

The third subcategory B.3 further highlights the students' actions but it also adds the goal-orient dimension to the actions.

"Studying is conscious and goal-oriented actions that aim at learning some thematic entity and/or skill. Studying has several sectors: seeking knowledge, searching the source of information (including oral guidance and feedback), voluntarily making exercises that relate to the topic and possibly partaking the organised pedagogical events. It might also include social aspects such as formal or informal group activities or discussions but this is not necessary (one can learn alone, too)."

The third main category, category C, focuses on a student. The first subcategory C.1 externalises studying as a certain part or time of the student's life. The second subcategory C.2 defines studying as a particular skill that a student possesses.

"Learn how to learn. Is able to analyse the topic that he/she does not know beforehand, focuses on substantial, find out what is essential and discard excess knowledge. In other words, has control over own time management and knows how to direct it to essentials. Studying teaches to think independently and to set, evaluate and achieve goals."

The third subcategory C.3 defines studying as a student's goal-oriented process that has a certain goal.

"Thinking and working process that aims at acquiring new knowledge, skills and attitudes. Lectures, literature and exercises may act as tools that are used in interaction with peers and teachers."

8.1.2 Teachers' conceptions of what kind of aspects are involved in studying

After the respondents had given their definition on what studying is they were asked to contemplate studying in more detail. The teachers were asked to consider *what kind of aspects are involved in studying*. The results suggested that the respondents focused on aspects that related to the study of computing. Some specified their response to specific paradigm or to an introductory/advanced course. The answers were read through several times and possible themes or aspects were listed along the process. The data analysis highlighted six main categories that emphasise different aspects: student characteristics and skills, theory and concepts, ability to apply the knowledge, in the studying environment, problem solving and the ability to program in a broader context.

The first category, the *student*, focused on the student's skills and qualities. The category consists of two subcategories. First, the *student's characteristics* subcategory emphasises qualities such as the ability to think logically and the ability to endure uncertainty. Second, the *student's skills* subcategory highlights aspects such as time managing, independents studying, social skills and ability to seek information.

The second category, the *knowledge*, emphasises the basics of computing related knowledge. The category divided into two subcategories, the *theory and concepts* and the *ability to apply the knowledge*. The subcategory theory and concepts focuses on *programming related structures and concepts* as well as on *understanding the basic idea of how computers work*. The second subcategory discusses the concepts and structures, yet the primary focus is on applying knowledge the students learned. The act of *practicing programming*, *learning proper procedures* and *learning to use design tools* are highlighted in this subcategory.

The third category, the *environment*, calls attention to the studying environment that is involved when studying. The ability to use the provided learning environments efficiently is underlined.

The fourth category focuses on *problem solving* and the process of learning as an important aspect of studying programming. This category in a way combines two earlier mentioned categories. The ability to solve problems calls for student to connect aspects, such as logical thinking and content-based aspects.

The last category, programming in a broader context, includes aspects such as the necessity of programming for future jobs, and knowledge of the application area. The summary of aspects of studying that were received as a result of the analysis is in Table 33. The subcategories that the most respondents mentioned were the ability to apply knowledge (22 respondents) and theory and concepts (21 respondents). The numbers of all mentions are listed later in Table 35, which summarises this and two other questions' answers.

Table 33 Aspects of studying

Main categories	Sub-categories /aspects	Extracts belonging to the category
Student	Student's characteristics	• Ability to think logically (6) • Endurance of uncertainty (1) • Motivation (1)
	Student's skills	• Ability to study independently (2) • Time managing (2) • Self-reflection and –evaluation (1) • Social skills (1) • Ability to seek for information (1)
Knowledge	Theory and concepts	• Concepts and structures (17) • Theory (5) • How computers work (6)
	Ability to apply the knowledge	• Applying the knowledge (4) • Reading and writing programs (3) • Practicing programming (12) • Learning to use the tools (1) • Learning to design, test, debug and document (12) • Understanding what happens when the program is running (1)
Environment	Studying environment	• Learning to use the studying environment (such as automated assessment tools) (3) • The use of the computer (while programming) (4)
Problem solving	Applying logical thinking into computing related problem solving	• Problem specification, analysis, solving and evaluation of the solution (6)
Programming in a broader context	Programming in context	• The purpose of programming (2) • Programming as a career (1) • Knowledge in the application area (1) • Project management (2) • The history of programming languages (1)

Note. Numbers in the parenthesis are the number of respondents who referred into a subcategory.

8.1.3 Teachers' conceptions of the focal aspects of studying and the difficulties students encounter

The respondents were asked to elaborate on which aspects of studying they thought to be essential when it comes to the successful studying. The categories that had unfolded in the earlier part of the analysis (Table 33) were used as a base for the analysis. The responses to the question "*Which aspects of studying play a central role when it comes to the successful studying*" were placed in the previously set categories. The categories referred by most respondents were: the student's characteristics (13 respondents), the ability to apply the knowledge (12 respondents) and the theory and concepts (10 respondents). Note that one respondent may have discussed several subcategories in her/his answer that belonged under one meta-level category/aspect. In Table 34, the numbers after each subcategory refer to the number of references each subcategory had. The respondents' answers focus varied a lot. Some focused purely on the programming language related features and others on student's characteristics as the following extracts illustrates.

> *"Understanding the basic programming concepts well. Absorbing at least the syntax, structure and concepts of a programming language – at least to a large degree."*

> *"1. Persistence, 2. Interest in intellectually interesting things, 3. The style that does not impugn too much, 4. Independent approach."*

In contrast, some subcategories that were mentioned as an aspect of studying (Table 33) were not regarded as focal aspects. For example, *the endurance of uncertainty, learning to use tools, the use of the computer while programming* and the most subcategories concerning *the programming in a broader context* were not referred as focal when it cames to successful studying. *Creativity* was the only new subcategory that emerged when the respondents were asked to list focal aspects in respect of successful studying. New extracts that belong to some subcategories are marked with the "+" in Table 34.

Subsequently, the respondents were asked to write about *the aspects of studying, which they observed the students had difficulties with*. The answers were then analysed in a manner similar to the previous question using the categories that had came up in previous phase. The responses confirmed the categorisation. Some new extracts belonging to sub-categories emerged and some categories did not unfold from this set of data. The extracts, *the endurance of uncertainty, the ability to independently study, social skills,* and all other extracts concerning *the programming in a broader context* did not receive any references hence they did not cause difficulties to the students. However, at the same time, several new extracts belonging to subcategories emerged. *Persistence, hard work, devoting to studying, IQ, exactness, level of abstraction and the shift from application concepts to programming concepts* were regarded as aspects that cause difficulties for the students.

> *"Laziness, the lack of practice, the difficulty to understand abstract concepts. The concepts are understood wrong and therefore the programs are built wrong. As a result there are poor and malfunctioning programs."*

The subcategories referred by most respondents as causing difficulties were the *ability to apply knowledge* (12 respondents), the *theory and concepts* (10 respondents). In addition, the *student's skills* (6 respondents) and *student's characteristics* (5 respondents) were reported. In a similar manner, respondent may have in her/his answer discussed several subcategories that belonged under one meta-level category/aspect and thus in Table 34 the numbers after each subcategory refer to the number of references each subcategory

Table 34 Computer science teachers' perceptions of focal and difficult aspects in studying

Main categories	Sub-categories /aspects	Focal aspect	Difficult aspect
Student	Student's characteristics	• Ability to think logically (8) • Motivation (6) + Creativity (1)	• Ability to think logically (3) • To be motivated (1) + To be persistence (2) + Overcome laziness (2) + Dedication (1) + IQ (1)
	Student's skills	• Ability to independently study (4) • Time managing (1) • Self-reflection and -evaluation (1) • Social skills (1) • Ability to seek for information (1)	• Time managing (3) • Self-reflection and – evaluation (1) • Ability to seek information (2) + Exactness (1)
Knowledge	Theory and concepts	• Concepts and structures (8) • Theory (1) • How computers work (2)	• Concepts and structures (8) • Theory (1) • How computers work (3) + Level of abstraction (3)
	Ability to apply the knowledge	• Applying the knowledge (3) • Reading and writing programs (2) • Practicing programming (5) • Learning to design, test, debug and document (2) • Understanding what happens when the program is run (1)	• Applying knowledge (1) • Reading and writing programs (4) • Practicing programming (3) • Learning to use the tools (1) • Learning to design, test, debug and document (3) • Understanding what happens when the program is run (2) + Shift from application's concepts to programming concepts (2)
Environment	Studying environment	• Learning to use the studying environment (such as automated assessment tools) (3)	• Learning to use studying environment (such as automated assessment tools) (3) • The use of the computer while programming (1)
Problem solving	Applying logical thinking into computing related problem solving	• Problem specification, analysis, solving and evaluation of the solution (3)	• Problem specification, analysis, solving and evaluation of the solution (2)
Programming in a broader context	Programming in context	• Project management (1)	

Note. Numbers in the parenthesis are the number of respondents who referred into a subcategory. New extracts that came out when the answers to the question at hand were analysed are marked with the "+" in the table.

Table 35 summarises the number of respondents who referred into subcategories during answering the three questions: What kind of aspects are involved in studying, which aspects are focal in respect to successful studying, and what kind of difficulties students encounter. Knowledge related aspects were referred as an aspect of studying by almost all respondents. However, when the focus is shifted on the aspects that are considered as having a focal role in successful studying, the student related aspects are highlighted, in addition to knowledge related aspects. When it comes to the difficulties the teachers have observed students to have, the focus turns back on the knowledge related aspects. The number of respondents is so small that it is not possible to make any strong conclusions based on the numerical data. However, the percentages in Table 35 may be interpreted as emerging trends that could be verified by further studies. The two highest percentages of each category are highlighted by bold type printing.

Table 35 Emerging possible trends

	aspects		focal		difficult	
	%	f	%	f	%	f
Student's characteristics	32	8	**52**	13	20	5
Student's skills	16	4	28	7	24	6
Theory & concepts	**84**	21	40	10	**40**	10
Ability to apply the knowledge	**88**	22	**48**	12	**48**	12
Learning environment	28	7	12	3	16	4
Problem solving	24	6	12	3	8	2
Programming in a broader context	20	5	4	1	0	0

8.1.4 Teachers' conceptions of how they can affect the students' studying process

The respondents were asked to consider *whether the teacher could affect the students studying process.* Six respondents were cautious about to what degree the teacher can affect students' studying. Student's low motivation or non-existent interest in studying was seen as issues that teachers could not affect. One respondent considered logical thinking and creativity as topics that cannot be taught. Bad habits were also seen as difficult to alter. The following example highlights the respondent's demure towards his/hers possibilities to affect students' studying process.

> *I have met some people who do not directly oppose learning but do not seem to have any real interest either (like I want the degree but not the skills and knowledge). When it comes to some difficult concepts I think you are reaching the limit what the teacher can and cannot do. Good illustrations and examples help to understand difficult concepts for some but sometimes it feels like the student's IQ is hindering learning.*

However, the majority of the respondents felt that they were able to affect the students' studying process. One respondent crystallised the positive attitude as follows.

> *The students need only the right attitude, willingness to learn accompanied with a touch of common sense and algorithmic thinking. The rest depends on the teaching personnel.*

The strategies the teachers' mentioned divided into two main categories: the *things teachers can do before the course starts* and *the things they can do during the course*. Both main categories consisted of several subcategories.

Things teachers can do before the course starts

The first subcategory concerns *choosing the content and modification of the knowledge* (five references). The teacher can choose such clear entities that the student is able to internalise them in the given lecture time. The teacher's job is to provide the knowledge in such a form that the students are able to receive it. This includes partitioning the knowledge into smaller entities, thinking about the way of presenting the knowledge and focusing on the focal aspects.

The second subcategory discusses the *teaching methods and pedagogical activities* (nine references). The teacher decides at the planning phase what kind of teaching methods he/she adopts, what kind of exercises students do, what are the assessment criteria and the ways of interaction that are used during the course. Thus, the teacher may affect the student's studying process in several ways. Pedagogical activities like group work and optional programming exercises coupled with deadlines were mentioned as a way to affect the studying process. The optional programming exercises were often mentioned as a good way to make students to work with determination and to get enough practice. Deadlines and bonus systems were regarded as a way to help students to divide their time in a reasonable way.

The last subcategory focuses on *the whole process*. The teacher needs to consider the whole process including all the details that might go wrong. If the course is well planned and there are no real reasons for dropping out of the course or studying altogether then the reasons lies on the student - for instance laziness or disinterest in the study field.

Things teachers can do during the course

The first subcategory focuses on *motivating students* (eight references). The motivating and encouragement were regarded as focal strategies to affect the studying process. One respondent described the importance of motivation as follows.

> *"The teaching personnel are in a focal role as a source of motivation. If the student is not motivated to start with, the teaching personnel have an opportunity enhance the motivation by positive attitude and by providing interesting courses and exercises. The same works the other way around, too. Motivation is easy to kill with uninteresting exercises or, for example, too difficult a course.*

The second subcategory concerns *teaching, guiding and explaining*. This category covers a wide range of issues from teaching students how to think to helping students with technical difficulties. Explaining concepts at lectures and providing alternative/new approaches are included into this category, too. However, the variation in students' knowledge level makes teaching challenging.

> *The tutor can affect the studying process by providing new viewpoints to the topic. However, this is challenging since you have to be able to communicate with many students at the same time. At this situation, it might be difficult to fulfil all requirements.*

The third subcategory discusses *the role of examples* (four references). The teacher can show, for instance, how to solve problems during the lectures. This provides the students with an example of how experienced programmers solve problems and how they think. The respondents mentioned providing examples and images helpful for students to understand difficult topics. However, the examples were seen beneficial also to pass on more abstract level skills like problem solving strategies.

Practical examples concerning the program coding and how you solve problems might be very revealing (e.g., the teacher writes code during the lecture). In general, to bring out examples and models how experienced programmer tackles problems. How does he/she think? It does not affect so much the learning of the language itself because it is about learning particular concepts and notation by heart

The fourth subcategory focuses on *the availability of the teacher to the students* (one reference). It was regarded important that the students were able to reach the teacher if needed. The technology provides ways to the student to communicate with the teacher if a face-to-face meeting is not possible.

The last subcategory concerns *the student's goals and studying process*. One respondent discussed the tutor's role as the one who gives the students clear goals and defines the process how to get to the goals.

8.1.5 Summary & analysis

The previous analysis showed that there are several strategies the teacher can use to affect the students' studying process. The comparison between the earlier listed difficulties (Table 34) and the strategies teachers may use show a positive connection. Next, some strategies to help students are mapped with the difficulties the students have according to the teachers (Table 36). Investing to motivate the students helps them to keep up the needed interest in the subject. Chosen teaching methods and pedagogical activities may help students to improve their time managing and self-reflection skills as well as the ability to seek information. Choosing clear entities for the course, applying appropriate teaching methods and taking time to explain and to show examples may help students with theory and difficult concepts as well learning how apply the knowledge learned. Showing examples of how experienced programmer solves problems may also be beneficial for the student who struggles with problem solving skills. However, some difficulties do not seem to pair with the strategies. For example, there were no mentions of strategies that could help with the issues like persistence and laziness. In addition, it is worthwhile to remember that six respondents were cautious to what extent the teacher can affect students' studying.

The previously discussed categories reveal several ways how the respondents perceive they can affect students' studying process. In this chapter, the categories the teachers mentioned are placed into the instructional process model that was discussed earlier in chapter five (section 5.2). Both the teachers' and the students' processes are discussed parallel to highlight connection between the two actors' processes.

During the planning phase, in the teacher's process, the teacher's decisions concerning the content and the way of presenting the content play an important role. The teacher may choose such entities that are reasonable for the student. Therefore, decisions like this affect the students' planning and studying phase. During the planning phase, the teacher can affect the students' planning phase, for example, by deciding the number and the extent of the programming exercises and setting deadlines for them. This information helps the students with their possible time managing difficulties. For instance, several smaller deadlines force the students further to divide the time they are allocating for a large, time-consuming project. During the planning phase, the teacher also decides what kinds of feedback forums are available for the student during the course. Naturally, the planned exercises and other pedagogical activities affect the great deal the students' studying phase by helping the students to understand difficult concepts and to get a practical experience in programming.

Table 36 Computer science teachers' strategies to help students with difficult aspects

Main categories	Sub-categories /aspects	Difficult aspect	Strategies to help
Student	Student's characteristics	• Ability to think logically (3) • Motivation (1) • Persistence (2) • Overcoming laziness (2) • Devotion to studying (1) • IQ (1)	• Motivating students
	Student's skills	• Time managing (3) • Self-reflection and –evaluation (1) • Ability to seek for information (2) • Exactness (1)	• Choosing appropriate teaching methods and pedagogical activities • Give students goals • Being available for students
Knowledge	Theory and concepts	• Concepts and structures (8) • Theory (1) • How computers work (3) • Level of abstraction (3)	• Choosing clear entities • Teaching methods and pedagogical activities • Explaining, teaching • Showing/giving examples • Being available for students
	Ability to apply the knowledge	• Applying the knowledge (1) • Reading and writing programs (4) • Practicing programming (3) • Learning to use the tools (1) • Learning to design, test, debug and document (3) • Understanding what happens when the program is run (2) • Shift from application's concepts to programming concepts (2)	• Choosing clear entities • Teaching methods and pedagogical activities • Explaining, teaching • Showing/giving examples • Being available for students
Environment	Studying environment	• Learning to use studying environment (like automated assessment tools) (3) • The use of the computer (while programming) (1)	• Choosing appropriate teaching methods and pedagogical activities
Problem solving	Applying logical thinking into computing related problem solving	• Problem specification, analysis, solving and evaluation of the solution (2)	• Showing/giving examples
Programming in a broader context	Programming in context		

Note. Numbers in the parenthesis are the number of respondents who referred into a subcategory.

The teaching phase in the teacher's process is in a central role when it comes to affecting students' studying process. First, the teacher or assistant teachers may help students to set goals for themselves as well as to define the studying process. Second, by motivating, implementing bonus systems and introducing optional exercises with deadlines the teacher can affect the student's planning phase. The different ways the teacher teaches, explains and gives examples affect the students' studying phase. The teacher's actions may help students to understand the content of the course and thus make the studying meaningful and rewarding. At the teaching phase, the teacher may also provide feedback to students, for instance, by grading exercises. The connection between the teacher's and the students' instructional processes are discussed in following paragraphs and visualised in Figure 39. The arrows between the student's and teacher's processes are one way arrows only because of the focus of this study is the teacher's conceptions of the studying process, and their affects on studying. In reality, students' process affects the teacher's process, too.

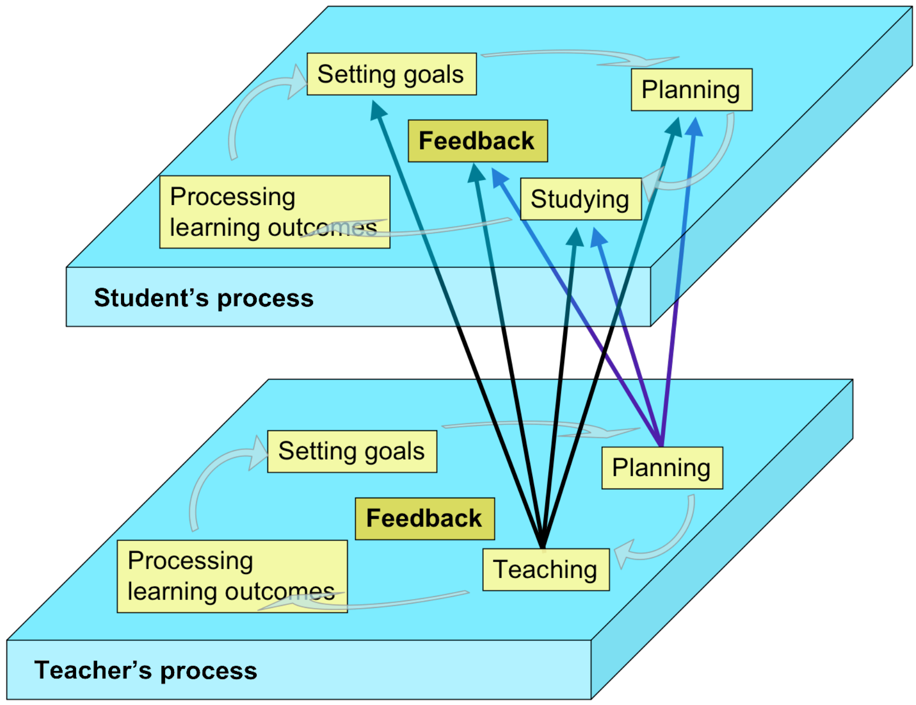


Figure 39 Teachers' possibilities to affect students' studying process

8.2 The teaching process from computer science teachers' point of view

The results of the CS teachers' questionnaire revealed what computer science teachers think about studying; which aspects are essential for successful studying, what aspects do students have difficulties with, and how the teacher can affect the studying process. The focus of the previous analysis was on teachers' thought about students' studying process; now the focus is shifted on what the teachers think about their own teaching process during the course. This part of the study aims at answering the following question: *How do the computer science teachers consider the different phases of the instructional process?*

Interviewees

Four teachers from the Helsinki University of Technology were interviewed during the academic year 2007/2008[18] (the interview plan is in Appendix 6). All interviews were held in Finnish, transcribed and translated into English by the researcher. In addition, notes were taken during the interviews. The teachers of the CS1 and CS2 courses were chosen as a target group since the focus of this thesis is on the instructional process in the field of computer science. At the Helsinki University of Technology (TKK), two teachers give large-scale CS1 courses for CS minors and CS majors. One teacher is involved with a small-scale CS1 course for the CS majors while the other teacher teachers the data structures and algorithms course (regarded here as CS2 course), which is a large-scale course for CS majors and CS minors.

The interviewee John has several years of experience in teaching. He has worked as an assistant teacher at several CS related courses. In addition, he has been teaching CS1 course for CS majors and CS minors for seven years. John has Master of Science degree in computer science and he is currently a postgraduate student. His research interest lies in the area of computing education.

The interviewee Kate has over 15 years working experience that consists largely of teaching and research at the Helsinki University of Technology. She has her Master degree in physics and her PhD in computer science. She has been teaching CS1 course for CS minors for over eight years.

The interviewee Alex has about ten years of teaching experience at this university; he has his PhD in computer science. Alex has been giving the data structures and algorithms course (regarded here as CS2 course) for CS majors and CS minors for over five years.

The interviewee Erik has over 20 years of working experience at the university with various teaching related positions. He has his PhD in computer science. Erik has been involved with a small-scale introductory programming course for CS majors that use problem-base learning for eight years.

8.2.1 Different phases of the instructional process from teachers' point of view

Setting goals phase

The goals setting phase included several aspects and several participants. The goals were discussed both on a degree program level and at a single course level. For example, Erik reveals the underlying goals of a degree program that has now operated for almost ten years.

> *Erik: I think that the reason why we wanted to establish such a degree program, at least that was one of the reasons, was that this computer science is such a male-dominated field. Well, it is characteristics for TKK [Helsinki University of Technology] anyway. But we felt that it would not be impossible to tempt also females to computer science if we just present it in a way that it does not appear so, how could I put it now, nerd –like… Then there was this opinion that multiply skilled persons are needed. The kind of people who in addition to computer science understands something about society, economy and people.*

[18] The interviewees agreed to participate to the study and that the results are published. The names of the interviewees have been changed.

Erik ...Instead, these other individual and group work related aspects. We have considered and discussed what the students who graduate from this degree program should be able to do and what they should be able to do already during their studies.

Another teacher formulated the general goal for CS majors as an ability to learn and an ability to communicate with the other professionals, teachers and researchers within this field. As a part of the other courses CS1/2 courses aims at that goal by introducing the basics.

Alex: ... Understanding the programming related concepts, acquiring the jargon so that you are able to communicate with other people within this field. Maybe you could even think that you could change the content of the course, the content is not so important. The ability to absorb new knowledge. If I think my own competence I think it is the reading and writing skills. Everything else is less important. In a way, after all, if you want to simplify things, it is the ability to read and write this field's literature. It starts from the basics like my course. Later on things just get a little bit more complicated… I see data structures and algorithms as a basic toolbox… I think it is one of the core courses. It is an essence of computer science, a central part of it.

The course goals were an end product of negotiations with other teachers, and the general university requirements. The challenge of the goals setting phase is meeting the demands of the departments, who require their students to learn programming. The kind of knowledge and skills of the computer science that other departments wish that their students would acquire sets some boundaries to the goals. On the other hand, when the CS majors are in focus, the challenge is the coherence and the sizing of the first and the second year studies in general.

Alex: This is like a constant small wave motion and change ... However, maybe it is good since one would think that at some point the courses click into places so that they are in a good relation with each others. That is probably the biggest challenge. I believe that each teacher is able to size his/her own course but who would be able to size the first and second year studies so that they are possible for a student to execute.

In a course level, this challenge appears in faculty discussions concerning the course content in relation to the other CS courses. Aspects like the kind of knowledge and skills students should have when they enter and leave the course are highlighted.

Alex: On a general level we, the teachers of CS1 and CS2 courses, discuss the goals every year… The goal setting starts at the meta-level, the course in relation to the other courses. Then we try to return to the course level. Certainly, the content of the courses stays the same for a long time. We make only small changes to them. We set the goals through the content. You have to master certain content in order to be able to enter the next course. In a same manner, we expect that students meet certain prerequisites. We have done this work together with the CS1 and CS2 teachers.

On the other hand, the teachers highlighted that there are also skills that should gradually evolve during several courses. Therefore, the collaboration with department's teaching faculty is in a central role when discussing on such goals and how to achieve them.

Alex: One thing what we have started to think is how we could encourage, highlight and even force the social grouping a little. So that they [students] would already be grouped when they come to the exercise groups. They would know each other's already and that would help them to form pairs [that are required by the teaching method chosen for the course]. So that they would not do those exercise alone. These are issues that an individual teacher has difficulties to influence to but if you are involved in departments' activities you may find a way... On the other hand, I do not have to be responsible for everything. There are other courses, too. This is a continuum where the same topics reaper many times….

The teacher's own opinions as a subject matter expert and his/hers background as well as the discussions with other CS1/2 teachers affected the goals of the course. The course content related goals were strongly affected by the ACM curriculum. The customs and assumptions also steered the content related goals. The CS1/2 teachers met regularly and discussed the content related goals in detail. In addition to the content related goals also more general type of skills were mentioned. Goals for CS minors and majors were also segregated to the degree. For example, the CS1 teacher who teaches only CS minors stated that the goal of the course is for students to feel confident in using computer programming to solve small problems that they face during their studies or duties. Erik takes the reasoning to level that is more concrete and discusses about working practice related goals.

> *Erik: Then there are goals that relate to the working practices both on an individual and group level. The purpose is that students learn how to seek and handle information. That is one of the focal ideas of problem-based learning. In addition we have them to do essays and mind maps. Handling knowledge and presenting knowledge. In addition to essays, they usually do portfolio, which we replaced with the group blog this year. We need to observe how we can develop it because it was not working in every way. To be able to handle information in every way. To be able to seek for information, to be able to analyse and to be able to do presentations of the information. Those are the individual level goals. In addition, the goal is that they learn group work skills…*

In addition, teachers also set goals for themselves concerning to improve the course, to plan effectively for the course, and implement their improvements in order to degrease the dropout rates.

> *John: If I think what kind of goals I have set for my course planning and implementing this year… I have taken some themes … such as increasing the students' activity during the lectures or increasing the transparency of the course arrangements. There are few rather abstract level goals…I will see after the course do I think those aspects have improved at this course.*

Feedback at the goal setting phase

There were several feedback sources for the teachers concerning the goals they have set. First, the teachers used their own background and knowledge to reflect on the suitability of the goals.

> *Kate: Well, as we reconstructed this course we discussed with the teachers from other departments. In addition, my background is not in computer science. Therefore, my experiences as a physicist and what for physicists need computer programming affect [the goals]. And although not all my students are physicists, I think that all masters of Science students need programming for similar things.*

Second, meetings with other CS1/2 teachers provided feedback.

> *Erik: ... Last summer the teachers of this degree programme that deal with Studio courses gathered together to consider the goals for these four courses. We also discussed what we should teach at Studio1 course, what the other teachers could expect the students to know when they enter other courses. How students' individual and group level knowledge evolve throughout the whole degree programme…*

Third, the department's discussion meetings provided a forum to deliberate the goals of the degree programme. In addition, discussions with other departments' personnel gave some feedback concerning the need of their students.

> *Kate: ... During the years I have come to the conclusion that object-oriented programming is not the best for my target group. The discussion with the other departments' teachers partly affected my opinion…. After that [discussion] I started thinking about it more… That is why we are going to change the language to Python.*

Adjusting measures

Adjusting the goals during the course was not possible in large-scale basic course since the course goals are set in relation to the other courses thus changing them would risk the coherence of the courses.

> *Alex: I do not think that I can change or adjust the goals much during the course anymore. I have published them at the first day. In addition, those goals have formed during several years and they are in relation to the other course's goals. They [other course's teachers] should be able to rely on that the goals stay for a long time… I could maybe change content related goals. Leave something out, some parts of a larger entity. You just find that you do not have time this year… And when it comes to these latent goals. The methods and forms of exercises are decided in good time. In a large-scale course… you cannot be very reactive… You have to be very proactive. Almost everything is decided before the course starts. Well, there is some room for adjustments.*

Communication with the students

The course goals are published on the course web pages. Especially, the content related goals were explained in detail. For example, the goals were divided into different levels; from minimum knowledge level through the level that one needs to master if one is planning to take advanced courses. Therefore, the student may choose the goals that fit the best his/hers interests and future plans.

> *John: I have three level goals. I have minimum goals that everybody needs to know. In a general level I say that one need to be able to read, write and rewrite smallish Java programs. Apply reading and writing skills into Java programming. That is the minimum goal. Then I have goals that go more into detail. They relate to the issues one needs at advanced courses. Which topics or skills one needs to posses at which level in order to succeed at advanced courses… Then I have a pile of bonus goals that you can try to achieve but you do not have to.*

Planning phase

The planning phase consisted of plans that explicitly state how the set goals could be achieved. However, the course history and past trivial experiences played an important role in the planning phase. Approved methods set some boundaries to the planning. For example, the programming exercises are generally thought to be the essential prerequisite for learning programming. Therefore, in the eyes of CS1 teachers the programming exercises played a central role.

> *John: ...There are a lot of things that are based on the history. It is generally accepted that programming exercises are in a central role in achieving any goals at the course. It does not help that somebody just lectures if students do not practice programming. There are a lot of such assumptions. In practice the planning, how you achieve the goals, ... the most important aspect of it is to plan the programming exercises so that they are related to the goals and, of course, other teaching methods, lectures and materials, too. Exam should relate to the goals too… I have tried to do explicit linkage between the exercises and goals.*

All the interviewed teachers had taught their course for several years, therefore, they did not need to start the planning from scratch. On contrary, the teachers took the previous year's course as a starting point. The general rule seemed to be that the teachers aimed at keeping the aspects of the course that worked well the last year and aimed at making small changes to improve the course. The cautious changing of the course plans was due to the large scale of the courses. It was too risky to change a lot at one time. If something went wrong in a large-scale course, there were several of hundreds of unhappy students.

Alex: The course starts so that we revise the previous course and see if there are some drawbacks. Then we start planning small changes. The starting point is to keep all that well working stuff. We do not start to plan from scratch the whole course. Instead, we make a small change, which is easy to take back next year if we notice that it does not work. This is the basic philosophy at the course, the starting point for planning.

The experiences from previous years moulded the plans for future courses. For example, one teacher explained how they had modified problem-based learning to suit the needs of the subject matter and to effectively steer students' studying habits.

Erik: ... In fact our way of giving the course has drifted further away from the problem-based learning... We anticipated already from the start that we could not build our course purely on problem-based learning. One has to also practice programming. It is after all a skill, not just knowledge. Programming exercises are a lot of work. Then there is this problem that students do not necessarily invest a lot time in PBL cases. We have tried to improve it with different means. During the last couple of years we have also discussed that it might be better that we would use a slightly different pattern for some of these meetings that what they used at the [other] university. We have now more design or constructive type of cases. We have couple of cases where they have to design the program in a group and then also to implement it... We have changed the nature of the cases a little bit. We cannot expect them to learn these topics through conventional PBL cases. Neither can we assume that they would do a lot of [independent] work related to the cases.

The implicit goals of the course became explicit at the planning phase; moreover, the relation between the goals and plans became visible. For example, one teacher wished to raise the pedagogical quality of the programming exercise meetings. Therefore, he decided to hire masters of Science level teachers and researchers as assistant teachers instead of settling for only undergraduate students. Another goal was to improve the course's passing rate by allocating attention to the timetables and students' use of time.

Alex: ... The basic group is four to six assistant teachers who run the programming exercise groups and grade the planning exercises. However, this year I wanted to upgrade it [the level of assistant teachers]... We have a culture that we hire second or third year students to assistant teachers. I wanted to upgrade that and I hired some researchers from my own research group to run the programming exercise groups. I thought it would be a way to create a real pedagogical situation to the programming exercise groups. So that it would not be just a situation where an assistant tries to control that you have made those exercises...

Erik: The goal is to contrive such people who are able to function in this kind of environments. You have to know something about social aspects, economic, society and computer science. That is the reason why group work and working in a group is characteristic for this degree program...

In general, the planning was described as an iterative process, which starts very early, even as early as a year before the course starts. Active actors at the planning phase were the teacher, the assistant teachers, and in one case, the researchers who were involved with the course related research project. However, the teacher was in the focal role.

Alex:... The idea is that I make a draft and then when we have meetings with the course personnel we go through them and develop them with each interest group. I have taken the whole gang to develop and plan together, especially to implement the plans but also to plan.

Erik: ...The last four years the head-assistant teachers have played an important role. They are students who have taken this course themselves and they have also worked as a normal assistant teacher. We have contemplated together with the head assistant teacher how to run this course. When we start to plan next year's course, we consider to which direction we should take it...

The planning phase included, for example, planning and making the programming exercises, written exam, content of the lectures, the web pages, along with other pedagogical material, and putting together a timetable for the course. Taking care of the practical arrangements like hiring assistant teachers was also included in the planning phase.

Feedback at the planning phase

The experiences of the previous years' served as an important source for feedback at the planning phase. New plans can be reflective of the previous experiences. For example, the teacher has some idea about which topics students would find difficult. Naturally, the course feedback that has been collected during several years gave some idea to how students thought about different course arrangements, timetable and the subject matter itself.

> *Erik: At the planning phase there is not so much feedback available. Mostly I and the head of assistant teachers and maybe the head of assistant teachers from the previous year think about it [the course]. We have used the feedback from the previous year's course. Based on the previous years' feedback we make the plans. It would be good to get some feedback concerning the plans. On the other hand, many of the things are such that it is difficult to foresee what follows from the change. You have to try it out to see what happens.*

The teachers disclosed that they frequently collect statistics during the course concerning, for example, the passing rates of the different parts of the course. These statistics then strongly steered the planning for the next year. The assistant teachers also contributed as a feedback source at the planning phase. Lastly, the researchers who are involved with the educational research project serve as a feedback source. One teacher discloses how the course related research has impacted the course planning.

> *Alex: ... They [research, results] often affect implicitly. At least it has affected the long-term goals. The one thing that I am going to do is that in future I will offer some self-access materials that include tutorial messages in addition to the TRAKLA assignments. In that way some of the lectures could be replaced with some other activity...*

Adjusting measures

The plans were mostly adjusted according to the feedback and experiences that were received from previous years. The goals of the course, however, remained mostly static. Interviewees expresses that they considered the coherence between the goals and the plans, yet, possible resulting changes were allocated to the plans and not to the goals.

> *Erik: ... It has been mainly on that level that we have considered these working methods. We have discussed about goals and what they [students] already are able to do. For example, earlier we taught them some HTML. Now people already know some of it or they can learn quickly... We had arranged meeting relating this artificial intelligence case where students came to introduce, it was a kind of oral presentation of their group work. This kind of considerations we have had concerning the goals and we have made the plans accordingly. But when it comes to the formal goals, those we have not [changed]. We have just pondered on how we should arrange the course so that we would achieve the goals.*

Communication with the students

At the beginning of the course, the teachers told the students how much time the course was likely to take. The course timetable, lecture plan and practical information were all placed at the course web pages.

Teaching phase

During the teaching phase, the plans are put into practice. However, the planning phase partly overlaps with the teaching phase. The teacher may still plan and implement some programming exercises during the course. At the teaching phase, the teacher gives lectures and may possibly run exercise groups. He/she reads and contributes to the course newsgroup discussion and is available for the students during the office hours. The teacher, the assistant teacher and researchers have regular meetings to discuss current course related issues.

> *Kate: At the implementation phase I give lectures and then I follow the newsgroup. I also answer quite a lot questions at the newsgroup. Every time when the students submit programming exercises to the Goblin [automated grading system] they can give feedback concerning the exercise. I follow that feedback and if I notice that the assignment is unclear I can still add something to the assignment. At the same time I see that the assistant teachers do their job and ... then I continue doing the programming exercises. Now, I am doing exercises to the make-up round.*

The teacher follows the feedback that is available from the automated grading system (verbal feedback and statistics) and adjusts the lectures and programming exercise assignments accordingly. He/she also gathers the statistics, like the number of students who completed the week's assignments. All teachers gathered experiences and ideas that they could use in following years. One interviewee, in particularly, made systematic notes concerning how the course arrangements worked and how new ideas could be used next year.

> *John: If it is possible to correct right the way I will do it. If not and I agree with the feedback ... I will make a note for the next year… for example, I read the course feedback and see how people in general have felt about it. Sometimes I have added some questions to the course feedback form based on the feedback I had during the course.*

Feedback at the teaching phase

The statistics proved to be an important source for feedback. Teachers compared this years' statistics to the last years' statistics and were therefore able to see if there were any major changes. The automated grading system, the statistics and verbal feedback it gathers, were all valuable feedback sources:

> *Alex: Maybe the most essential quantitative feedback that I gather myself is the number of students who use the systems, how many exercises have been made …Then I compare it to the previous year. If there are big differences then I need to start to ask why… There are several sources for feedback signals. I try to process those signals and interpret where we are going. This is like an ocean liner. It does not turn very quickly. You need to use small manoeuvres and you cannot jump up and down.*

The discussion in the course newsgroup and the online discussion forum gave some feedback as well. Likewise, students' questions during the lectures or lecture intermissions, lecture feedback notes (a short description of unclear issues/questions that students submitted after each lecture) and the feedback form at the end of the course were all valuable information sources for the teachers. Meetings with the assistant teachers gave feedback concerning the difficult material for the students.

> *John: For example, there are many good questions at the lecture feedback forms, which I could include in my next year teaching… It [lecture related feedback from the students] affects the next years' course planning ... it affect more to the teaching method than to the goals. It might affect the goals too. I suppose it is a form of feedback… when students fill in the question how much time they used for an exercise. From there we get the statistics… it affects crucially next years' goals. I think that next year we will partly lower the level of difficulty…*

Adjusting measures/possibilities

Small adjustments were possible to make to the timetable, lectures and exercises assignments if the feedback suggested them. For example, if the previous programming exercise proved to be difficult for the students the teacher may have repeated some topics. However, the programming exercises themselves are more difficult to adjust since they need to be finalised four weeks before they are published.

> *Kate: I might do very little changes to the next lecture if I notice that some topics are not clear… I have adjusted lectures but not the goals [during the course]…*

Even though there are limited possibilities to make changes at the large-scale courses, there were exceptions based on individual preferences among the teachers. For instance, some preferred to have everything finalised before the course begins, while others were ready to make small adjustments along the way.

> *Alex: You can adjust the timetables to some degree. I did not publish all the exercise assignments at once. Therefore, I was able to adjust the workload in a middle of the course, to choose assignments according to what students had learned. That kind of adjustments. However, you have to follow through the basic idea. At the end of the spring when we have the results then we deliberate is this worthwhile to repeat next year.*

> *John: You cannot change the big lines. In general, I do not like to change them in a middle of the course. It is my personal decision. I try to inform the students as much as possible as early as possible. Everything should be as ready as it can be by the time the course starts and everybody knows how this course proceeds. Especially as this is a large-scale course. Otherwise, there would be too much confusion. It would have to be an emergency before I would change the structure of exercise rounds or which exercises are compulsory, the stuff that affects the goals.*

In a small-scale course, it was possible to make changes to the teaching and studying methods even during the course.

> *Erik: We have not changed the goals during the same course. Teaching and studying methods we have altered to some degree but that happened more during the first years. We had some PBL cases that did not work very well and because of that we had to change the teaching methods so that we gave a lecture about those topics. But that did not happen often. It has more or less been such that we have adjusted the grading guidelines during the course. It is explicitly so that you gather the feedback during the course and then you use it when you plan the next course.*

Communication with the students

The teacher gives lectures and possibly runs the programming exercise groups. During the lectures, the teacher might give hints to which topics are important and remind the students of the goals. Lecture slides and other educational material are available for the students. Programming assignments clearly state which content and skills are necessary. The teacher or the assistant teachers are available for the students during the lectures, programming exercise group meetings and via the newsgroup discussion on-line.

> *John: I have tried to bring them up [goals] during the course, to remind them that now would be a good time to deliberate your own goals, there are materials where you can check what you could learn during this course…*

Processing learning outcomes phase

At the processing learning outcomes phase, the results of the instructional process start to emerge. The interviewees listed several different types of outcomes of their instruction, such as grades and students who have learned something. In addition, statistics, process description of the whole instructional process and the feedback from the different phases of the process were regarded as an outcome at the instructional process. Moreover, there are outcomes that start to come up already during the course like grades from the exercise rounds. On the other hand, there are outcomes that are available only after the whole course is over – sometimes even a long time after the course is over.

John: Outcomes. There are of course, the grades and the passing rate… the description of the whole course, how the whole process went and how did the students cope. Actually, all course arrangements and experiences are a sort of outcome because they set a starting point for next years' course. That is how I think about it. Then we have course feedback, all sorts of summaries and feedback in different forms. We got feedback forms, feedback from the exercises and feedback from the lectures, IRC logs, which are a sort of feedback, too… My own notes concerning the course are an outcome, too.

The students' submissions provided the teachers a closer and more specific view of what students had learned and how the goals had been achieved.

Alex: I have read some of the submitted planning exercises. I have tried to see what they reveal. Communication and directions have been understood… I have also graded at least one or two of the smaller written exams. So after the course [when grading the exams] I get a pretty good feedback of how effective the teaching was…

Kate: ...programming exercises and exams… I can observe from them whether the students have learned what they were supposed to.

Feedback at the outcome phase

The sources that helped to interpret the outcomes were statistics, grades, the results from the course feedback form and other feedback systems, such as the teachers own notes and own self-reflections. Statistics were mentioned as the most salient and effective feedback form. In a large-scale course, it is not possible to follow individual students. Instead, you have to observe the big trends.

Alex: What I monitor is the quantity of the students who pass the exercise rounds. That is an important number. And it gets smaller as the course proceeds. From that slope, I try to predict how well this course is going. I am not concerned about the quality. Because the students who get a grade, they are not a problem. I am mostly concerned about how many people I can keep on the course. That is the most salient number I follow. You have to observe mass, you cannot follow individuals.

On the other hand, there were less feedback sources when the courses were discussed in a larger context, like the course as a part of a degree program. For example, the signals from former graduated students working in industry do not reach individual teacher. Therefore, the teachers own activity in seeking the feedback was essential.

Alex: There is no feedback [from the working life] that would come straight to the teacher, not even if you would try to dig it. I know that data structures and algorithms as a subject matter is extremely esteemed in the eyes of employers and even employees. Some studies have been made where the recently graduated persons have evaluated the course. The data structures and algorithms course seems to be in top three.

In addition, the teachers' own observations played a central role in receiving the feedback.

Erik: The best feedback in the long-term I have gained when I have followed how students act here later. It seems that they are able to use these methods, they can work in a group… I followed them [a small group of students] and their outputs. They used similar working methods and it seemed to work well. At the first course, these learning diaries and portfolios are still a little bit faltering. But later on it really seems like that they would have become working methods. They were really able to use these methods.

Erik: One guy is involved in an algorithm research concerning data mining. I talked to him and he told that they have applications that concern natural science. He said that this education has been helpful for him in situations where he has to discuss and explain things to people who have a quite different background.

Adjusting measures

The outcomes of the course served as an important feedback source for other phases during the instructional process. However, in a large-scale course it was often not possible to utilise the feedback during the same course.

John: How they [students] keep up. I cannot… say that I would utilise that information during the course. I do not have the time. We have deadlines every week. I get only that kind of feedback that this exercise round was difficult and a part of the students did not make it. But at that point I cannot fix it anymore even though it was too difficult… I can try to help those who managed to pass that difficult exercise round by clarifying some topics at the lecture… I cannot help the students who flunk the exercise round anymore…

Kate: The statistics concerning the programming exercise round come late in a way. At the moment, the programming exercise rounds have deadlines for the topics that have been taught two week ago. You cannot use time for that anymore. But then next time [when the course is arranged].

Communication with the students

The students get points and grades as a formal expression of learning. During the course, the automated grading system gives some feedback to the student concerning their submissions. At the end of the course teachers arrange the open events when the students can come and revise their exams.

Kate: Goblin [automated grading system] gives feedback from the programming exercises. We have tried to improve that feedback. Now it tells what kind of things we test. Therefore, if student's program fails some particular test then he/she can consider that his/hers program did not work correctly. It [automated grading system] does not tell the student which skill he/she is missing but it tells what kind of tasks his/her program is not able to perform…

Problematic issues concerning the degree program/larger entity

During the interviews, some problematic issues concerning the organisation and degree program level issues emerged. In relation to the studies, one teacher brought up that the degree programs themselves are not coherent. Especially, the basic studies modules are fragmented and it is difficult to perceive how some courses relate to the entity as a whole.

Erik: I think the one of the biggest problems in this degree program is that there is no coherence… If you want to get an overall picture … you ask from a person who has just graduated or is about to graduate. The one who is smart and has been an active student, someone who has worked as a assistant teacher at different courses. They are usually the ones who have the best picture of this [degree programme]. There is no more coherent view available. There are many reasons why degree programs are so fragmental. One reason is that basic mathematics and physics are a large portion of many degree programs studies… Consequently, at the beginning of the studies the goals and content of the courses are outsourced from the degree program's point of view. Math department teaches what it teaches…. This contributes to the fact that we are not aware of what happens at which course. At the major studies level there might be some discussion on how these courses relate to each other.

Relating to the previous extract, the flow of information between teachers was regarded partial. Some teachers were not informed what was taught at other courses and felt that there was no proper forum for that kind of discussions.

> *Erik: …We do not have any other forum where we would really exchange information. What are the courses' content and requirements or what is required from previous courses? Sometimes there are vague discussions.*

8.2.2 Summary of the phases from the teachers' point of view

In summary, the content played a major role in the setting goals phase; the goals were determined by the content. Customs and the way previous courses had been arranged steered strongly the planning and implementing of the courses. At the teaching phase, the plans were implemented and only small adjustments were possible. Different kinds of outcomes were mentioned. The feedback was gathered during the process and it was utilised during the next instructional process. There was a variety of feedback that teachers received during the process. The scale of the course was connected to the type of the feedback that was gathered and determined as informative. In large-scale courses, the statistics were a focal feedback form, whereas in small-scale courses the feedback was more qualitative in nature.

> *Erik: ... We do not gather feedback with feedback forms but we look at their portfolios and see what kind of feedback we get. There we get qualitative feedback and not just quantitative feedback and that is useful when we develop this course. On the other hand, we have feedback from degree program committee and in general, from the people who have something to do with this degree program, the teachers of the other courses. They have considered this as a good course… It is not obvious that this small group would have their own programming course. However, the course has established its own status pretty quickly… It is regarded an important course. It always has positive feedback.*

There were also various ways of how the feedback was utilised. On the one hand, the feedback resulted in small changes to the courses and, on the other hand, perceived trends in the field could result generation of a new course or even a new degree program.

> *Erik: ... The information technology has become service centred… The application is not anymore coherent piece where there are different classes and other modules. Instead, we can think that we have separate services and these services can use other services. Then on the other hand, you should think about these issues from the users' point of view. This kind of service centred way of thinking is a strong trend… Now we are founding the international master degree to this field…*

The interviews highlight that developing the course is an iterative process that takes several iteration rounds. In addition, since there are many aspects that affect the studying/teaching process and the outcomes, it is not always clear how some modifications to the course affect the studying and teaching processes. In that sense, a course can be seen as a loosely coupled system where cause-effect relations are not clearly visible. The following extract is from the teacher who has hired some researchers[19] from his own research group to work as an assistant teacher at the course where traditionally all assistant teacher had been undergraduate students.

> *Interviewer: Last time we met you told that you had hired researchers as assistant teachers to your course. How did those labs with researcher assistant teachers go?*

[19] Researchers' own field of study was related to CSER and thus they were familiar with the educational literature, at least to some degree.

Alex: Well, it is difficult to say since I was not present but I have made statistics and compared the outcomes with two last years. The difference is that in previous years there have been some 30% of students who have not been able to get any grade from the exercises. They will drop out of the course at some point. This percent has dropped to about 23%. ... There are of course several aspects that contribute to it. Supposedly, the type of guidance they got… Then on the one hand I ... explicitly reserved time from students' calendars. That is, this year there were exercise groups more often and at the same time the exercises got a bit more difficult, too. This might have resulted that people go more motivated. There was more challenge. Anyway, they got better grades. As I said there are several factors here. Assistant teachers are one factor.

To conclude, several aspects affect the teaching process. Traditional organisation of the course, the scale of the course, and course's relation to the other courses set some basic boundaries to the course. Those boundaries together with the practical arrangements form a big fraction of the teaching process (Figure 40).

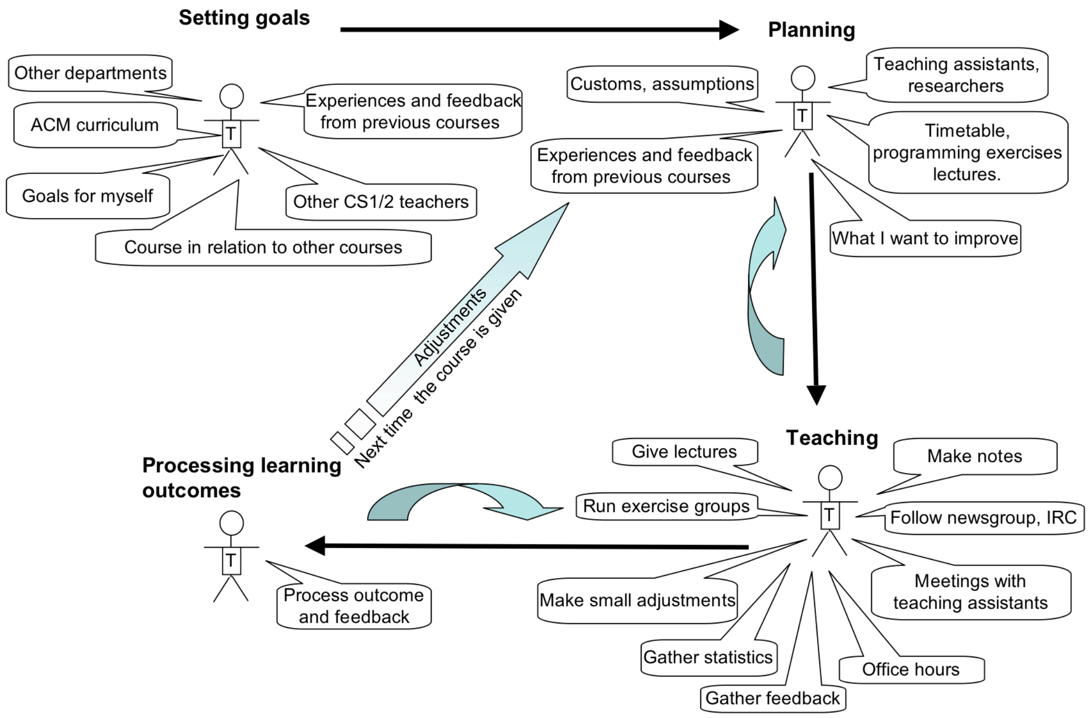


Figure 40 The instructional process through the teachers' eyes

8.2.3 Analysis & discussion

The teacher's teaching process is filled with various aspects that affect their decisions and actions. First, there are many different collaborators in both the goal setting phase as well as in the planning and teaching phases. Second, some boundaries limit the teacher. Therefore, the teacher is by no means free to implement the course in a way he/she feels is necessary. There are several issues the teacher needs to consider, such as how the course is going to fit in with other courses, are the goals of the course in line with other courses' goals that precede it, what are the customs to teach the course like this in this university and what are the resources for the course. Therefore, on one hand, the teacher's scope of actions is limited. On the other hand, the boundaries and customs ease the implementation of the course. Especially, if the course has been running successfully for several years, the teacher does not have a need to do big changes to the process. This saves time and effort that could be used, for instance, in refining smaller details of the course.

The results suggested that there are several different types of feedbacks available during the process; the student feedback forms are only a small proportion. The teacher's active role in receiving the feedback is essential. The feedback forms that students fill in at the end of the course are the only feedback source that the department organises. If the teacher wants some other feedback, he/she has to seek that by him/herself. Particularly, if the teachers would like to have feedback from sources that are not directly connected to the students at the course (e.g., assistant teachers) they have to be active and obtain feedback by themselves. One teacher reported that he observed fourth year students, and interviewed recently graduated student who just started working at the university. This gave him an insight that the studying methods that were used throughout the degree program as indeed useful tools for the students. The other teacher pointed out that getting feedback from the graduated working students is not easy. Such feedback could be gathered, but it rarely reaches an individual teacher. In summary, the amount and the variety of feedback, that the teacher can use to revise his/hers instructional process, may vary considerably from teacher to teacher.

Based on the interview data, the students at the course and the teaching organisation affect the teachers' instructional process in many ways (Figure 41). The CS teachers receive feedback from the students' studying phase. For instance, the time the students used for doing the programming assignments affected the teachers' setting goals phase the next time the course was given. Based on that feedback, some small revisions were made to the goals. The feedback from the students' studying phase affected also the teachers' teaching phase; particularly, it affected the teaching methods teachers' decided to use. The students' learning outcomes served the teachers a lot of useful feedback. Statistics concerning the passing rates and grades were the feedback the teachers used during planning, teaching and processing learning outcomes phases. However, often this feedback was not utilised until the course was given again next year. The teaching organisation affected the teachers' instructional process by organising meetings where several teachers together discussed the goals and plans of the larger study entity. For example, the meetings where the representatives of other departments were able to express their expectations concerning the level of programming competence they would like their students to receive, affected the teachers' setting goals phase.

Finally, the utilisation of the feedback is not straightforward. For instance, in large-scale courses the teacher makes only small changes every time the course is given. The standard is that one keeps everything that works and improves aspects that do not work. Since, the development of the course is a slow process, it needs several iterations. As there are several aspects that play a part in the teaching and studying processes, it is not always easy to be sure, what causes which observed outcome(s). Using general system theory terms, the course may be defined as a loosely coupled open system. A characteristic of the loosely coupled systems is that one cannot predict precisely what kind of outputs the system is going to produce. As an open system, several aspects affect the instructional process that the teacher cannot fully control. In other words, there are issues that the teachers can affect, for instance, by changing the approach to teaching from lecture based teaching to problem-based learning. On the other hand, there are issues that are out of the teachers' control. For instance, several issues affect the student's learning outcomes. The student might have unexpected family commitments that required his time and thus result in the decision to drop the course. On the other hand, when the student finds out that there is an opening and a good opportunity for jobs he/she becomes very motivated in learning the subject matter. Thus at times, the efforts the teachers does to try to enhance the students' learning could seem to be highly effective or ineffective because teachers cannot control all external factors that influence student success in their course. The more realistic conclusion is that the teacher may not be able to identify exactly to which degree the new tool or pedagogical approach was involved with improved learning outcomes.

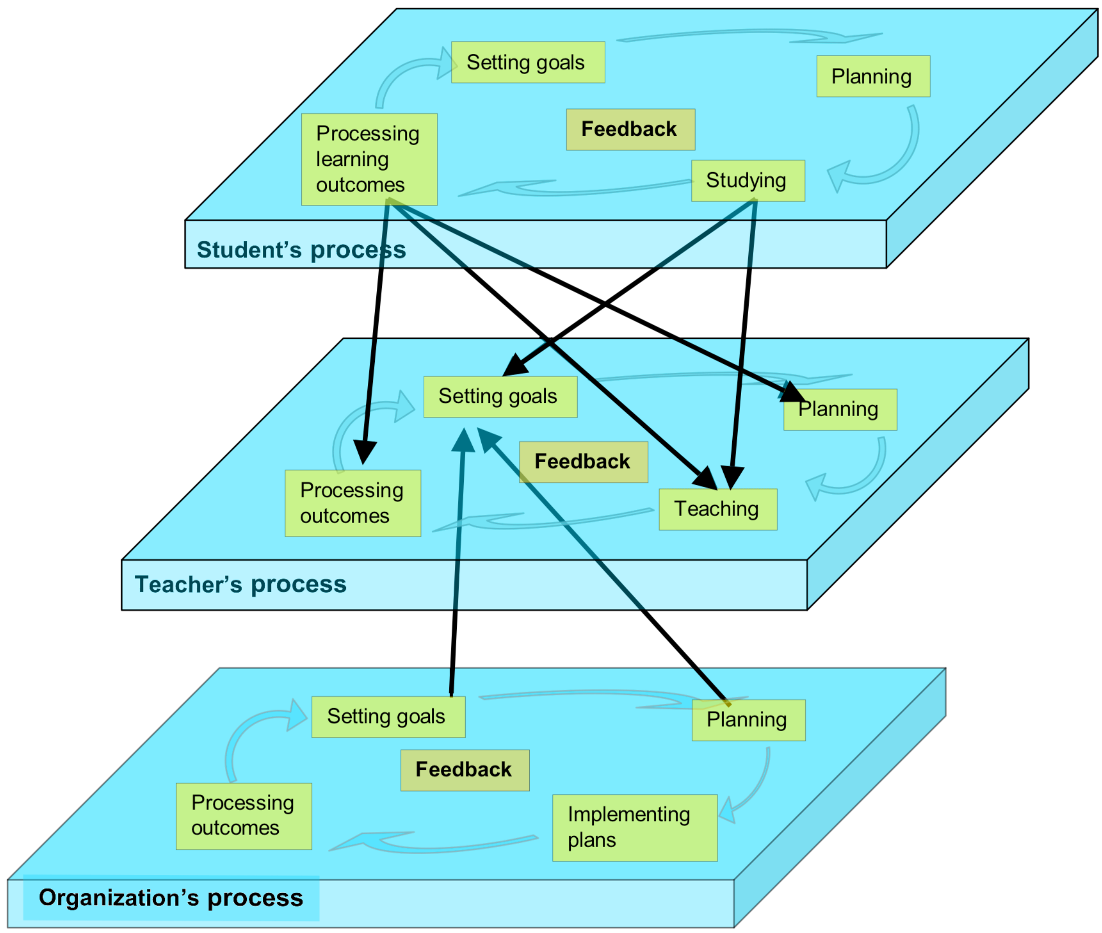


Figure 41 Students' and teaching organisation's ways of affecting the teachers' instructional process

8.3 Summary of the instructional process from teachers' point of view

The studied CS teachers' definition of studying generally divided into three main categories: action, student, and knowledge and skill. The most often mentioned definitions concerned the action of studying; studying is hard work, collecting knowledge, and accomplishing goal-oriented actions. The student category identified; studying as a phase in a student's life, as a student's skill, and as a student's goal-oriented mental process. The knowledge and skill category concerned knowledge: building up knowledge, acquiring new knowledge, and acquiring knowledge with a clear goal in mind.

These definitions of studying reflect the teachers' views on aspects of studying. Five main aspects came from the data: student, knowledge, environment, problem solving and programming in a larger context. Most respondents mentioned knowledge related issues as aspects of studying.

The teachers' perception of which aspects are essential for successful studying included student related aspects as well as knowledge related aspects. However, when the teachers were asked to elaborate on aspects that students have difficulties with, the focus shifted back on learning and applying computer science theory and concepts.

The teachers' perceptions varied concerning the degree to which their efforts affect the students' studying process. Some respondents were suspicious of the degree they could affect the individual student's studying process. However, most respondents reported at least one strategy to promote studying. The strategies varied from motivating students to applying appropriate teaching methods, and providing pedagogical activities that would help the students learn. In summary, the strategies the teacher mentioned affected many phases in the students' studying process.

The CS teachers' interviews revealed many aspects that affect the teacher's teaching process. The interviews highlighted several factors and boundaries that set some requirements for the teacher and, on the other hand, helped the planning and delivery of the course. The customs steered the teaching processes. The courses were organised the way they had been organised the year before and only small adjustments were made to the procedures. The goals of the course were often inherited from previous years' courses and were not thought over explicitly every year. However, the teachers were conscious of different kinds of goals. Some interviewees made a difference between the goals they set for the students and the goals they set for them self. The planning phase was also strongly affected by conventions. The planning of the course was described as an iterative process. Only small changes were made throughout the year. Several different feedback sources also became visible from the data. For instance, a teacher's own reflections and experiences from previous years, statistics on grades and drop-out rates served as feedback for the teachers. The possibilities and challenges to get and utilise feedback were identified. In a large-scale course, the teachers were not able to utilise all the feedback immediately. Instead, the feedback was used when the course was given the next time. In a small-scale course, the teachers were able to adjust the ongoing course according to the feedback they received.

9 The instructional process from the teaching organisation's point of view

The two previous chapters presented the instructional process from the students' and the teachers' point of views. The third research question, which is the core of this chapter, *How is the instructional process seen at the organisation level* focuses on the teaching organisation's point of view on the instructional process. The question was answered with a two-part research project. The first part concentrated on answering the research question 3.1 *How are the different phases of the instructional process and the feedback about the process and outcomes presented in documents?* The formal documents, such as the Universities Act and the guidelines the teaching organisation gave, were used as data source. The analysis highlighted the education related main themes found on the different levels of documents. In addition, the review of the formal resolutions of the Rector and the Dean highlighted the independent role of the Faculties and Departments. The second part of the study focused on answering the question: *How do representatives of the administration see the instructional process?* The analysis revealed several issues that the interviewees found problematic in the instructional process. The description on the research procedure and the discussion of quality of the procedure is in chapter 6, section 6.3.

9.1 The instructional process through formal documents

Helsinki University of Technology (TKK) has recently gone through a comprehensive organisational change. During the change, the number of administrative units was radically reduced. Before the organisational change, the university had 12 departments (hereafter referred to as "old departments"), which consisted of laboratories. The first of January 2008, the 12 old departments merged into four faculties. The faculties consist of new departments. Due to this change, some of the older faculty and department level documents did not correspond directly to the documents of the new organisation. However, in the analysis, there was no need to make a clear distinction between the content of the faculty and the department level documents and therefore the lack of correspondence is not a problem.

Several formal documents define goals and plans for the university. For the analysis I selected documents at three different levels: at the society level, at Helsinki University of Technology level, and at faculty and department level. At the faculty level, I focused on the documents that concerned the Faculty of Information and Natural Sciences. In addition, some Department of Computer Science and Engineering level documents were included in the analysis. However, it was not suitable for the purpose of this study to separate the faculty and department level documents. Often the goals and plans were stated at the faculty level documents, but the department was the executive level. The general criterion for the selected documents was that they stated something about teaching and studying. Documents and statements that focused more on the administration of the university were left outside of this analysis.

Table 37 includes the list of documents that were used in the content analysis. From the faculty and the department level documents the faculty's performance agreement 2007-2009, the faculty's study guide and the quality handbook were written during the old organisation. The data that was chosen for this content analysis discussed mainly the basic studies or the quality of teaching and education. Therefore, this analysis highlighted only extracts of formal documents.

Table 37 Documents included in the analysis

	Documents included in the analysis
Society	• Universities Act (645/1997 + changes)[20] • Government Decree on University Degrees[21]
Helsinki University of Technology (TKK)	• Performance agreement between TKK and the Ministry of Education[22] • Degree Regulation of the Helsinki University of Technology[23] • Action handbook of the Helsinki University of Technology [24]
Faculty and Department	• Faculty's performance agreement 2007-2009 and 2008-2010[25] • Faculty's handbook[26] • Faculty's study guide[27] • Quality handbook[28] • Department's handbook[29]

During the analysis process, the sentences that discussed the goals, plans, teaching/studying and outcomes were highlighted. In addition, preference was given to the regulations that concerned the basic studies or could be interpreted to concern also basic studies. This selection was based on the focus of this thesis, which is the instructional process at an introductory level course.

The highlighted extracts were placed into tables where goals, plans, implementing plans and outcomes were displayed side by side. In the first round of analysis, the focus was on the different topics that were expressed in the different level documents and whether the documents emphasised the continuous process from goals to plans, and further to

[20] Yliopistolaki 27.6.1997/645
[21] Valtioneuvoston asetus yliopistojen tutkinnoista 794/2004
[22] Opetusministeriön ja Teknillisen korkeakoulun välinen tulossopimus kaudelle 2007-2009 ja voimavarat vuodelle 2007. Opetusministeriö 10.1.2007. Retrieved on March 2, 2008, from www.minedu.fi/export/sites/default/OPM/Koulutus/yliopistokoulutus/hallinto_ohjaus_ja_rahoitus/yliopistojen_tulossopimukset/tulossopimukset_2009/Teknillinen_korkeakoulu.pdf.
[23] Teknillisen korkeakoulun tutkintosääntö. Hyväksytty Teknillisen korkeakoulun hallituksessa 13 päivänä joulukuuta 2004. (muutokset 9.3.2009 mukaan lukien). Retrieved on April 3, 2009, from http://www.tkk.fi/fi/opinnot/opintohallinto/tutkintosaanto/tutkintosaanto.pdf
[24] Toimintakäsikirja versio 1.2. 2008. Teknillinen Korkeakoulu. Retrieved on April 25, 2009 from http://laatu.tkk.fi/fi/toimintakasikirja/toimintakasikirja_ver_1.2.pdf
[25] Teknillinen korkeakoulu, tietotekniikka. Tulossopimus 2007-09 ja 2008-10. Retrieved on April 14, 2009, from http://unel.hut.fi/pls/wwwunel/wwwunel.wwv_main.main?p_cornerid=3404&p_currcornerid=43&p_language=sf&p_edit=0&p_full=1&p_cornertype=item&p_iscornerlink=1
[26] Informaatio- ja luonnontieteiden tiedekunta.Toimintakäsikirja, versio 0.003. 2009. Teknillinen Korkeakoulu. Retrieved on April 25, 2009 from http://laatu.tkk.fi/fi/toimintakasikirja/il_tdkn_toimintakasikirja_14_1_2009.pdf
[27] Tietotekniikan tutkinto-ohjelman opinto-opas 2007-2008. Retrieved on April 14, 2009 from http://www.tkk.fi/Yksikot/Osastot/T/Tieto/Opinnot/oppaat.html
[28] Laatukäsikirja. Teknillinen korkeakoulu, Tietotekniikan osasto. 7.9.2007. Retrieved on January 10, 2008, from http://intra.cs.hut.fi/laatu/
[29] Informaatio-ja luonnontieteiden tiedekunta. Tietotekniikan laitos. Laitoskäsikirja. Versio 0.007. 2008 Retrieved on April 14, 2009 from http://laatu.tkk.fi/fi/toimintakasikirja/il_t_laitoskasikirja_081112.pdf

the implementation of the plans and their outcomes. In the second round of analysis, the focus was on how the themes found in the different level documents were presented in other level documents.

Society level documents

Three main goals were identified when the Universities Act and Government Decree on University Degrees were analysed. These documents gave guidelines for all universities in Finland, on a general level. The three goals were: 1) universities aim at providing higher education that is based on research, 2) universities aim at achieving a high quality of research, education and teaching, 3) general goal statements concerning the knowledge and skills that the students should be provided with. The main themes from this study's point of view were research based higher education and the high quality of research, education and teaching.

The plans to achieve the goals were expressed the Universities Act and Government Decree on University Degrees in the form of a rough structure of studies leading to a university degree. The structure that was stated in the society level documents highlighted that the studies may include, for instance, basic, intermediate and advanced studies, language and communication studies, interdisciplinary studies, other studies and internships. Since the plans were on a general level, they allowed each university the possibility to design the structure of their own degree programs according to their needs and wishes. These society level documents did not state anything about the subject matter content of the studies.

The teaching and studying phase was identified in these documents as guidelines for how many hours of work are expected from full-time students per one academic year. The documents also stated the time limits for the universities concerning the education they provide. The universities must arrange the education so that the student can complete the lower university degree (BSc, BA) in three years and the higher university degree (MSc, MA) in additional two years.

The learning outcomes were presented in these documents as statements concerning the students' ability to demonstrate that they have attained the objectives of the degree. The summary of the statements in the society level documents is presented in Table 38.

Table 38 Summary of the main themes in the society level documents

Goals	Plans	Implementing plans	Outcomes
Universities act, Government Decree on University Degrees (2004): Higher education based on research **Universities act:** A high international quality of research, education and teaching **Government Decree on University Degrees (2004):** The education shall provide the student with (section 12): (1) Good overall knowledge of the major subject or a corresponding entity and conversance with the fundamentals of the minor subject or good knowledge of the advanced studies included in the degree programme (2) Knowledge and skills needed to apply scientific knowledge and scientific methods or knowledge and skills needed for independent and demanding artistic work (3) Knowledge and skills needed for independently operating as an expert and developer of the field (4) Knowledge and skills needed for scientific or artistic postgraduate education (5) Good language and communication skills.	**Government Decree on University Degrees (2004):** The studies leading to the higher university degree may include (section 15): (1) Basic and intermediate studies and advanced studies (2) Language and communication studies (3) Interdisciplinary study programmes; (4) Other studies (5) Internship in order to improving expertise.	**Government Decree on University Degrees (2004):** The average input of 1600 working hours needed for studies of one academic year shall correspond to 60 credits. (section 6) The university must arrange the education to enable the student to complete the [lower] degree in three years of full-time study. (section 8) The university must arrange the education to enable the student to complete the [higher] degree in two years of full-time studies. (section 8)	**Government Decree on University Degrees (2004):** The student must demonstrate that he/she has attained the objectives set for the degree, studies, thesis, and language proficiency (section 16)

University level documents

The university level documents that were included in the analysis concerned the entire Helsinki University of Technology (TKK). These documents did not focus on teaching and education provided by one particular Faculty or Department. Instead, the statements concerning the goals, plans, teaching/studying and outcomes concerned all Faculties and Departments at the Helsinki University of Technology.

The main goals that were stated already at the society level documents were also seen at the university level. However, in the university level documents, such statements were more focused. The goal of the Helsinki University of Technology is to provide higher education based on research in technology and architecture and to promote scientific and artistic education. The second goal was to provide high quality research and teaching. The third goal concerned the knowledge and skills that the education should provide the students with. At the university level documents, some of the knowledge and skills statements were more precise than in the society level documents. For example, in the society level documents, one of the goals was to provide knowledge and skills in order to work as an expert and developer of the chosen field. In the university level documents, the same statement was made in more concrete terms (knowledge and skills to understand the problems of the field from the point of view of users, of technical and social systems as well as of the environment). In addition to the goals that were inherited from the society level documents, two new goal statements came from the university level documents. The first statement concerned the shorter completion time for studies. This statement related to the perceived problem of prolonged studentship. The second statement concerned the knowledge and skills for continuing and flexible learning.

The plans to achieve high quality teaching and education that is based on research included actions such as strengthening the international co-operation and development of the quality systems and paying attention to the quality of the education and teaching systems. The plans to achieve the knowledge and skills related goals included a structure for the studies. In the university level documents, the structure was more precise than in the society level documents. The university level documents demonstrated the content of the studies at a general level and the extent of the study modules.

In the university level documents, several statements concerned how a shorter completion time for studies could be achieved. The plans included actions, such as the development of the quality of the student admission procedure and teaching, as well as the planning, guidance and follow-up of the studies. In addition, the faculty planned to provide study counselling, promote students' well-being, and provide each student with a tutor teacher.

The actions to implement the high quality of education and teaching included the general statement that the university develops its teaching and evaluation methods continuously. The statements concerning actual teaching focused on the needed work input per students per one academic year (1600 hours), the structure of the academic year (four teaching periods) and the forms in which teaching is provided (e.g., lectures, exercises and seminars).

The statements concerning the outcomes reflected the society level statement that the students must demonstrate that they have attained the objectives set for the degree. In addition, these statements focused also on the grading system by stating the meaning of the different grades: 5 = excellent, 4 = very good, 3 = good, 2 = very satisfactory, 1 = satisfactory. The university level documents also brought forward some new outcomes that the university can use to evaluate the effectiveness of its actions. These outcomes included, for instance, course feedback on the course content and the studying process, statistics on the number and the quality of degrees, and the competence of the teachers. The summary of the statements in the university level documents is presented in Table 39.

Table 39 Summary of the main themes in the university level documents

Goals	Plans	Implementing plans	Outcomes
Performance agreement (2007) between TKK and the Ministry of Education, Action Handbook of the TKK: The goal of the university is a high quality of research, postgraduate education, teaching, and artistic activity. The goal for Finnish universities is to be competitive partners especially in the area of European higher education and research. **Performance agreement (2007) between TKK and the Ministry of Education:** Shortening the completion time of the studies at the university. **Action Handbook of the TKK:** TKK emphasises higher university degree level education. The goal is to promote the students' active role. The goal is to bind the research and the teaching tighter together. Curriculum forms a coordinated entity. The content and the process of teaching are constantly developed. Teaching is high-quality, interesting, current and reacts to the changing needs of employers, society and science. **Degree Regulation of the Helsinki University of Technology (2004):** To provide higher education based on research in technology and architecture, to promote scientific and artistic education also	**Performance agreement (2007) between TKK and the Ministry of Education:** … strengthening the international co-operation and by developing the quality system. The quality of student admission and steering system will be made more effective. The quality of the teaching, planning, guidance and follow-up of the studies will be developed so that the completion time of the studies becomes shorter. The student and working life feedback systems will be developed. The university plans the degree programmes so that they … shorten the time used for studies and reduce the needless exchange of degree programmes. Universities … pay special attention to the quality of education and teaching methods. Universities formulate a strategy for life-long learning. Universities promote … students' well-being and development of students' study skills. **Action Handbook of the TKK:** Learning environments, teaching methods, international co-operation in education and teachers' pedagogical training will be developed. The teaching and studying culture will be renewed. Teaching and learning are highlighted in addition to research.	**Degree Regulation of the Helsinki University of Technology (2004):** The measure of the extent of studies is a credit unit. Courses are quantified credits according to the workload required. The average input of 1600 working hours needed for studies of one academic year correspond to 60 credits. (4 §) Teaching of the University is provided in the form of courses. Courses of a degree program are compulsory, alternative or elective. (6 §) Teaching provided by the University is scheduled in teaching periods. There are four teaching periods in an academic year. Teaching at the University is given in the form of lectures, exercises, seminars, study excursions and other applicable methods. The University develops its teaching and evaluation methods continuously. (55§) Students studying for lower and higher university degrees have the right to study in accordance with their personal study plan. (56 §)	**Performance agreement (2007) between TKK and the Ministry of Education:** The realization of the goals stated in the performance agreement will be estimated yearly during the performance agreement negotiations. **Degree Regulation of the Helsinki University of Technology (2004):** In lower, higher and postgraduate degrees the grades used for accepted study attainments are Excellent (5), very good (4), good (3), very satisfactory (2) and satisfactory (1). In addition, for well-founded reasons, accepted study attainments of some courses can also be given the grade 'pass'. (60 §) To be awarded a Master's degree, a student must complete the studies. A student must demonstrate that he/she has attained the objectives set for the degree, studies and a Master's thesis and the language proficiency (23 §) **Action Handbook of the TKK:** The uniform course feedback concerning the content and the

otherwise, and to educate students to serve their country and humanity. (1 §)			process.
Degree Regulation of the Helsinki University of Technology (2004): 1) Good overall knowledge of a major subject of the degree programme; 2) Knowledge and skills needed to apply scientific knowledge and scientific methods or prerequisites needed for independent and demanding artistic work and knowledge and skills for continuing and flexible learning; 3) Knowledge and skills to understand the problems of the field, from the point of view of users, technical and social systems as well as environment; 4) Knowledge and skills to operate as an expert and developer of the field in working life; 5) Good language and communication skills 6) Knowledge and skills for scientific or artistic postgraduate education. Education shall be based on scientific research or artistic activity as well as practices in the field. (23 §)	**Degree Regulation of the Helsinki University of Technology (2004):** Each student studying for a lower or higher degree shall have a tutor teacher to guide him/her in his/her studies… departments also provide other study counselling. Each student … shall have a personal study plan (8 §). **Degree Regulation of the Helsinki University of Technology (2004):** Studies leading to a Master's degree include: 1) Studies of methodological principles (10 credits); 2) Three modules, at least one of which shall be an advanced module of a major subject of the student's own degree programme. Only one of the modules can be a basic module. (20 + 20 + 20 credits); 3) Elective studies (at least 20 credits), as well as 4) A Master's thesis (30 credits). (May 2, 2005) Subject studies and advanced studies are included in the modules. (23 §)		Statistics about the number and the quality of degrees: e.g., the drop-out rates and the division of grades. How graduates are employed: alumni, statistics, surveys. Formal competence of the teachers: sufficient teacher student ratio, teachers' scientific and pedagogical competence, rewards.

Faculty and department level documents

In the faculty and the department level documents, some of the goal statements were inherited from the higher-level documents. The goals concerning research based higher education and the high quality of research, education and teaching were seen in the faculty and department level documents. However, the analysis of the faculty and department level documents revealed several new goal statements that were not seen in the society or the university level documents. The goal statements at the faculty and department level were more concrete and related to the scientific field than the goal statements in the university or in the society level documents.

The goals that were inherited from the university level documents concerned the high quality of research, education and teaching. In the faculty and department level documents, the focus was especially on a high quality basic education. The other goals derived from the university level documents concerned the shorter completion time for studies. The new goal statements, which were introduced for the first time at the faculty and department level, included themes such as: responding to the changing challenges in the industry and information society, and producing of well-educated experts for the national and international labour markets.

The plans to achieve the goals varied according to the goals. For example, in order to be able to respond to the changing challenges of the industry the faculty and department does internationally valued information technology related research, and teaches experts capable of tackling challenging assignments, who are ethically responsible and cooperative. The plans concerning the production of experts for the needs for working life focused on the evaluation of the instruction and gathering feedback from working life.

The goal of recruiting the best students was planned to be achieved by marketing the degree programmes to high-school students and by enhancing the international good reputation of the faculty through the English masters programmes. The plans concerning the shorter completion time for studies included actions such as focusing on the study guidance and encouraging students to commit to their major. The plans to provide high quality basic studies focused on the assessment of the education, encouraging teachers to take part in pedagogical training and national and international co-operation and comparisons.

The teaching and studying phases were expressed as sentences stating that the departments and teachers were responsible for organising their courses and exams. The faculty's task is to yearly publish a degree programme study guide, which contains the structure of a degree and the list of courses that were included into the study programme.

The themes that belonged to the learning outcome phase included feedback from graduates, alumni, and employers, appreciation of the study degree, outcomes of the courses, and the amount of years needed to complete a degree. The faculty and department used the feedback from the courses, graduates, alumni and employers as an outcome. The appreciation of the degree was concretised as the number and the quality of the student applications the faculty received. The outcomes of courses included: the passing rates and the average grades of the courses. The summary of the faculty level statements concerning the goals, plans, teaching/studying, and learning outcomes is presented in Table 40.

Table 40 Summary of the main themes in the faculty and department level documents

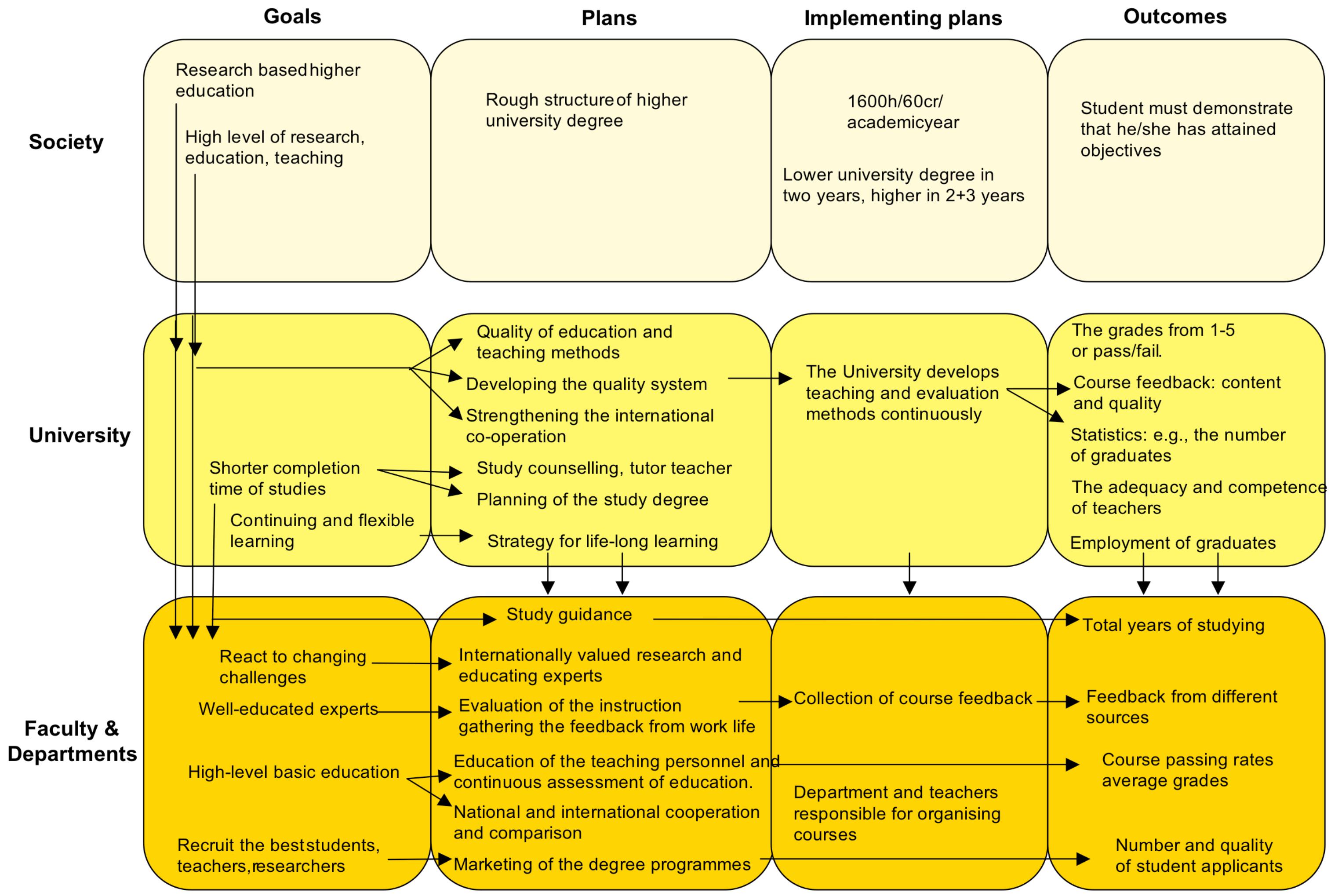
Goals	Plans	Implementing plans	Outcomes
Faculty's performance agreement 2007-09 and Faculty's study guide[30]: The Faculty of Information and Natural Sciences is internationally appreciated as the nation's leading teaching and research unit. The education that the faculty provides gives an ability to develop and apply information technology. It also gives ability to master entities in many related technical and other fields of society. The basic task of the Faculty is to react to the changing challenges of industrial life and information society. Faculty's central goal is to produce well-educated labour to meet the needs of the society and with the help of research and education to create a level of expertise that firms can use in innovation and product development. **Faculty's performance agreement 2007-09:** The goal is to educate competitive experts for the national and international labour markets and scientific communities. The strategic goal of the Faculty is to: - To recruit and motivate the best students, researchers and teachers of the field - To provide high-quality basic education	**Faculty's performance agreement 2007-09 and Faculty's study guide:** …carrying on internationally valued information technology related research and educating experts who are able to tackle challenging assignments, are ethically responsible and cooperative. **Faculty's performance agreement 2007-09:** To increase international good reputation with the help of English Masters programmes Degree programmes have new brochures and www pages that are directed to high-school graduates. The Faculty has a team that focuses on marketing degree programmes to high school graduates …national and international co-operation and comparison. Focusing on the faculty's education and continuous assessment of education. Widening the evaluation process of the instruction; especially gathering the feedback from working life. … focusing on study guidance and by encouraging students to commit to their major subject. Faculty proposes a project to intensify study guidance for students whose	**Faculty's quality handbook:** The departments are responsible for organising the teaching. Departments and the teacher in charge are responsible for organising the course and the exams. Faculty publishes yearly degree programme's study guides that include degree requirements and other practicalities that students need to know. Faculty collects feedback systematically from different levels. Course feedback is collected automatically from all courses. Evaluation team discusses the feedback and quantitative feedback summary. The feedback with the summary is sent to departments. At the faculty level the summary of course feedback and the quality of faculty's education is done yearly.	**Faculty's quality handbook:** The number of graduates vs. accepted students The total years of studying Feedback from graduates and alumni Feedback from employers Course feedback Appreciation (e.g., the number and the quality of student applicants) The passing rate of the courses The average grades of the courses **Faculty's handbook:** The quality of the basic education: course feedback The teacher-student ratio

[30] Note that these documents were written before the organisation change. The documents are closer to current department level documents than the faculty level documents.

- To increase international good reputation with the help of English Masters programmes The goal is that the degrees that the faculty grants are of high quality. The Faculty aims at preventing prolonged studentship. The goal is to improve the quality of basic studies and to revise the content of the studies. **Department's handbook:** The Departments basic task it to do scientific research, teaching and having an influence on society according to the guidelines the Faculty has given.	studies have been delayed. Teachers are encouraged to take part in pedagogical education and information and communication technologies education. The co-operation between teachers will be strengthen by organising internal training. Assistant teachers have also a possibility to give feedback on their work and the course they worked on. Faculty considers the measures to increase response rate of the student feedback surveys and how to utilise the feedback better. The faculty's study guide includes the structure and the content of the studies, e.g., the list of the courses that are included in different majors in the CSE Degree Program.		

Summing up, all the phases of the instructional process described in chapter five were represented in the documents. The statements became more concrete as the level of the documents came closer to where the goals and plans were implemented. The goals and especially the plans were listed in a more concrete manner for the first time at the faculty and department level. The two main education quality related goals (research based higher education and the high quality of research, education and teaching) were visible at all levels. The statements concerning how the goals would be achieved (the plans and actual teaching/studying actions) and statements concerning the outcomes were the most plentiful in the faculty and department level documents. Table 41 summarises the instructional process related statements that were found in the documents. The table does not contain all the themes that were found in the documents. Instead, priority was given to the statements that concerned the quality of the education the teaching organisation provides. The arrows in the table stand for the connections between the goals, plans and outcomes.

Table 41 Summary of the main instructional process related statements found in the documents



9.2 The teaching organisation's guidelines concerning the instructional process

The formal documents that were analysed in the previous section revealed some main themes concerning the goals, plans, teaching/studying and outcomes. This sector describes what kind of guidelines the organisation gives concerning the education related actions. I have taken the formal resolutions that central persons in the teaching organisation give, as an indication of the way the organisation directs the actions and the decision-making in the teaching organisation. Figure 42 summarises Helsinki University of Technology's (TKK) organisation and highlights the central persons who make decisions in the organisation. The Figure 42 is simplified and many committees that help the Rector, the Deans, and the Department heads to make their decisions are left out of the figure. This decision was made to highlight the central persons and their role in the teaching organisation.

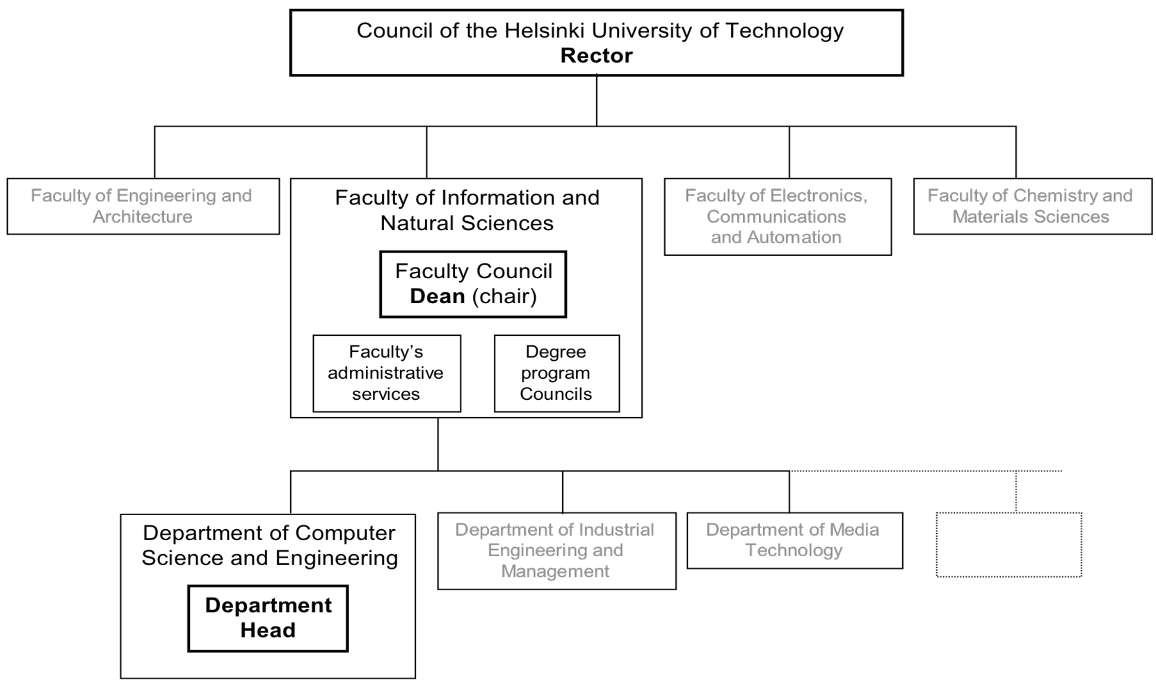


Figure 42 Central persons in Helsinki University of Technology's (TKK) organisation

The Council is the highest decision-making body of TKK. The Rector is the chair of the Council. Other members are Vice Rectors, the Deans of the four faculties and the representatives of employees and students. The Council has several tasks that are related to both the education the University provides and the University's economy. Examples of education related tasks are decision making on the establishment and discontinuation of degree programmes as well as on their aims. The Rector, as the chair of the Council, has a central role in the organisation. She or he has several tasks that relate to the managing, supervising and developing of the activities of the University. For instance, the Rector is responsible for admitting new students.

The Rector also gives resolutions that affect the whole University. During the years 2005 – 2009, the Rector has given altogether 33 resolutions. The statements in resolutions are often on a general level since they give guidelines for the whole University. For example, in the resolution concerning the enhancement of the fluency of the studies[31] the Rector gives guidelines for the actions to develop the degree programs. The resolution states, for instance, who is responsible for the development work (faculties), and what tasks must be accomplished (e.g., learning outcomes should be defined for degree programs, majors, study modules, and courses). The resolution mentions also the importance of education development work. Many other Rector's resolutions concern the teaching organisation. For instance, the resolution on degree program's development[32], defines the role of the Faculty Council, Dean, Degree programs Councils and other people who are responsible for the content, quality and organising of education.

The highest decision-making body of a faculty is the Faculty Council. The Faculty Council consists of the Dean, the Department Heads, representatives of staff and students and two non-university members. The Dean is the chair of the Faculty Council and thus directs the faculty according to the Faculty Council's decisions. The duties of the Dean consist of directing, developing and monitoring the activities of the faculty and taking responsibility for their effectiveness, allocating the resources of the faculty and issuing the diplomas for bachelor's degrees. The Dean may also give resolutions to Department. Since the new organisation came into effect on January first 2008, the Dean of the Faculty of Information and Natural Sciences has given only one resolution. This resolution stated the responsibilities of the professors who are responsible for degree programs. The tasks were stated in a general manner. For instance, the professors are responsible for relevant preparation of the degree-level curriculum.

A Department Head directs each department. The Department Head's tasks include: to direct, develop and monitor the activities of the department. The Department Head may also decide how the use the money that has been allocated to the department.

In summary, based on the formal resolutions, the teaching organisation gives only general guidelines how to develop and implement teaching. The teaching organisation gives mainly general frameworks for the faculties and departments concerning who is responsible for what. Therefore, faculties and departments are relatively free to organise their actions the way they want.

9.3 The instructional process from the point of view of representatives of the administration

The aim of this chapter is to answer the question: *How do representatives of the administration see the instructional process?* Three representatives of Helsinki University of Technology (TKK) organisation were interviewed in Finnish during the academic year 2007-2008 (the interview plan is in Appendix 7). All interviews were held in Finnish and they were recorded and transcribed by the researcher. In addition, notes were taken during the interview. The quotes below are authors' translations from Finnish transcription.

Interviewees were a planning officer, a professor and a training manager. The planning officer's responsibilities included the quality check of the teaching and the

[31] Opintojen sujuvuuden edistäminen. Toimenpide hallituksen päätökseen 9.2.2009. Retrieved on April 23, 2009 from http://www.tkk.fi/fi/opinnot/opintohallinto/paatokset/20090330opintojensujuvuus.pdf

[32] Tutkinto-ohjelman kehittämisen ja päätöksenteon vastuutoimijat tiedekunnissa. Retrieved on April 23, 2009, from http://www.tkk.fi/fi/opinnot/opintohallinto/paatokset/20090220tdkvastuutoimijat.pdf

administration of postgraduate studies in the Department of Computer Science and Engineering. The quality check consists of the assessment and developing measures of the teaching of the computer science degree program. The planning officer had a Master of Education degree and had been working at TKK for over seven years, in different positions. The planning officer represents the purely administrative aspect of the instructional process.

The professor from the Department of Computer Science and Engineering had over 20 years working experience at TKK; he had been appointed to a professor's post for eight years. His responsibilities consisted mainly of three tasks: teaching, research and administration. For example, the professor attended several working groups and committees at different organisation levels within TKK in addition to teaching at courses/seminars and being an active member in the international community of computing education researchers. The professor's interview represents the view that combines the administrative aspect with the teacher's point of view.

The training manager had a wide educational background. In addition to natural science, her background included information specialist studies. At the time of the interview, she was doing her postgraduate studies at TKK. She had over 20 years working life experience and had been working at TKK over 10 years. Her responsibilities consist of several national and international projects that concerned the developing of education, especially technology enhanced learning. She also teaches within further education. The training manager's interview represents a more general point of view to the organisation.

All three interviewees discussed the instructional process from a different point of view. The training manager was concerned about the coherence of the studies and how the degree program level goals were in relation to the modules and further to the individual courses. The planning officer related the faculty level goals to her concrete assignments, such as evaluation of the quality of education. The interview with the professor highlighted the gap between the formal goals and the actual planning process that took place in committees. In the following section, the instructional processes and especially the challenging or insufficiently working aspects of the process are highlighted.

The instructional process

Several problematic issues came forward when the representatives of the administration discussed the *setting goals* and *planning* phases. The interviewees emphasised that the goal statements in the formal documents are at a high abstraction level and that there is no strong connection between the formal goals and the actual concrete actions. The high level of abstraction resulted in some challenging duties for the planning officer. She needed to reify and operationalise the goals to observe the quality of the education.

> *Planning officer: "To educate experts in the field of information technology, who are able to take on challenging tasks, are ethically responsible and cooperative"'. These I should then reify and cut into smaller entities. Then there is a strategic goal "... to recruit the best teachers and students, high quality education". I try to observe the high quality education. Or it [high quality education] should be expressed with it [course feedback]. These are tricky issues since the goals are written like this.*

The other problematic issues concerning the *setting goals* and *planning* phases relate to the prevailing culture at the department. The culture promoted the substance-first approach to the planning of studies. The larger goals concerning the skills and knowledge entities were not regarded as a starting point for the *setting goals* and *planning*. In addition, the goals of the courses were not always stated clearly or the quality of the statements varied. It seemed that the prevailing culture of the planning affected also the content and the quality of the course level goal statements.

> *Planning officer: ... This is exactly the same problem I talked about earlier. Here the planning starts at course level. We have these 100 courses. Now, how can we include them into a degree? Not so that we would instead start at the goals concerning what the students should know and what skills they should posses. This whole culture is based on courses… The word curriculum is not understood here. Degree requirements define the courses that are needed for a degree. Degree requirements mean that there are these modules and they have this content. But the curriculum covers a broader range of issues, it defines the goals of the degree. Here we have never made the curriculum. All we make are degree requirements… That is why, for example, degree programs do not have official goals.*

The same notion concerning the substance-first approach to planning came up during the professor's interview. However, the professor was more concerned about the balance of the substance and the soft skills. In addition, the professor emphasised that there was no organised system, which would make sure that adequate soft skills would be taught during the studies. He also proposed that the lack of an organised planning system was partly due to the inadequate goal setting.

> *Professor: Our planning focuses too much on the substance, on the content. Then there are skills, that students should possess. This includes, on the one hand, technical skills and, on the other hand, soft skills. How soft skills are part of the planning – that is not co-ordinated. You cannot teach soft skills in one course… we should think about these soft skills when we decide on our teaching methods. However, we do not have a proper method to match our teaching methods so that, e.g., during the basic studies there would be some number of group works, essays, or oral presentations. What we have done has been ad hoc type of actions… there is no systematic procedure for this in our planning process.*

The third problematic issue was that the teachers of the department did not know about the goals that were stated in the formal documents. For example, the performance agreement concerns all employees and gives them the rough guidelines what to do and what they should be striving for. However, this was not the case at the Department.

> *Planning officer: ... Everybody plans his or her own little things. This is said now very ugly but that is what I think…. This is not only our department's problem. This is quite a common problem. Maybe they see this thing differently at enterprises. Enterprises have a very strong goal. Everybody knows what the goal is, what they are striving for and they act accordingly. But do we know here what our goal is? If we think that we have a goal, a plan, an implementation and then we evaluate the outcomes. But how can this cycle work if we do not know what our goal is? They [teachers] do not know what the department has promised to the rector and what department's goals are. I am very negative now, I know, but I think this is how it is for the most part. Certainly there are some teachers who know, but if we select randomly a teacher, does (s)he know? On what do they base their courses' goals and content?*

The interview with the professor highlighted another aspect to how the role of the formal goals was seen during the planning processes. According to the professor, the actual planning was often steered by conventions, and hence practical problems arose from the course level issues and timetable issues. For example, there was a concrete problem at the department level: many students did not advance in their studies the way they should have. Therefore, a team was established to find out the reasons for that and to plan proper interventions. However, it was perceived difficult to follow how the studies proceeded because, for instance, there were no statistics available concerning the progression of studies.

Further notions on the planning included the unclear goals. The goals of the planning process were often not clearly stated nor discussed during the *planning* phase. The exception was the year 2005 when the structure of the degree was changed. During that year, the coherence of the courses was discussed profoundly. However, since that the old mode of setting goals had taken back its place. The plans were made according to the previous year's plans and changes were made only if there was a clear need for them.

Professor: The setting goals process is incremental by nature. We have a starting point, which is the instruction we provide now. Let us assume that this present instruction is working, its content and implementation is working. Changes are made only when well-defined problems or deficiencies unfold or when some new technology comes that needs to be included into teaching. Then we react and make plans how these changes could be matched. If we already have plans and they have worked well in the past, then we update only 10% of the plans and not the entity …. The goals are in a way implicit and tacit knowledge. Therefore, planning is focused on needs for change.

The last problematic issue concerning the planning of the instruction was that it did not focus on larger entities. For example, the degree as an entity was not discussed. Rather, the discussions concerned smaller entities and practical problems.

Feedback concerning the instructional process

The interviews brought forward several sources from which the representatives of the administration received feedback. The departments and faculties got feedback from the higher organisation level during the performance agreement discussions. The meaningful purpose of the goals and the content of the studies, as well as the teaching methods, were tested against the feedback received from the graduates. Other feedback sources were their own contacts to the enterprises/industry, the research done in the department, the trade unions and the discussions with the other departments' representatives. However, the procedures to collect the feedback from different sources were not always systematic and organised. Even though the different feedback sources did not have a formal role in the planning phase, important aspects of the feedback were discussed during the planning.

Professor: ... Occasionally we make inquires to the industry but there is no organised system for that so that we would do it, for instance, every three years. Rather they are inspired by a temporary need. And of course, the department's professors and researchers have plenty of contacts with the industry and enterprises. That is a way to receive feedback concerning whether they think our students are able to do the right things. This feedback comes indirectly. Even though we do not ask for it explicitly it might come forward during discussions.

The interviews highlighted several problematic issues concerning the feedback. First, the feedback is not always collected. Both the professor and the training manager brought forward that the larger study entity level feedback is not collected. At the course level, the feedback is collected after each course. The department also collects feedback from the graduates. However, no feedback is collected from the study module level. Second, the collected feedback is not utilised well. The feedback might be available but it is not utilised during the *setting goals* or the *planning* phases.

Professor: ... Changes are made to the module. The content of the course is changed… teachers teach their courses and then there emerges outcomes… The outcomes are not used as feedback at the planning phase. This feedback is usually not used. It is not actively sought for. I have an image that when changes are made for the module… the presumption is that students learn these things. There is no attempt to verify that otherwise. The starting point is that what has been planned, that is what students learn. If some deficiencies come forward then some corrections are made.

Planning officer: I have made the summary of the course feedback with the comments for all departments… then I have sent it to the departments and asked them to comment on them. Only one Department has responded, others may have discussed the feedback or then not.

The training manager highlighted more general types of problems at the organisation level. These issues concerned the whole university rather than some specific faculty or department. She perceived the feedback as a multi-layer loop. There are smaller feedback loops and then there are larger loops that contain the feedback that is received from the society. She highlighted the university's problem to collect and utilise the feedback. For instance, the university did not provide enough feedback channels for the employees. The training manager perceived this as problematic, especially, since the university is about to merge with two other universities. At times of big changes it would be important for everybody who is affected by the change to be able to give feedback, and thus have an opportunity to influence the outcome.

> *Training manager: TKK does not have any feedback channels for employees. Sure, there are all sorts of email addresses but they do not tempt much. For example, this innovation university. They ask all employees. That is not enough. There has to be other channels too. I think we are handicapped when it comes to our own organisation. We are innovative when it comes to many other things but not when it comes to our own organisation.*

According to the training manager, the consequence of not having the feedback channels for employees was that there was a danger that TKK was not able to locate and use experts within its own organisation.

> *Training manager: Here some individual researcher might be a lot better expert than the Dean of the faculty. If you only look at the hierarchical organisation, you may loose expertise. There is expertise available but you do not use it. This is a matter of running the university.*

9.4 Summary of the instructional process from the teaching organisation's point of view

The analysis of formal documents highlighted several themes. In addition to the general goals of high-quality research, education, teaching and research-based higher education, there were other more concrete goals and plans in the university, faculty and department level documents. For example, the themes concerning the shorter completion time of studies[33] and recruitment of the best students, teachers and researchers came forward in the faculty and department level documents. The review of the formal resolutions of the Rector of Helsinki University of Technology, the Dean of the Faculty of Information and Natural Sciences, and the Head of Department suggests that the faculty and department are free to act reasonably independently. The Rector and the Dean do not give strict guidelines for how faculties or departments should arrange their functions. This observation is in line with the culture of academic freedom in universities. Traditionally, researchers and teachers are free to arrange their work the way they want. However, since researchers and teachers are a part of a larger organisation, which has its own general goals, there may be tension between the requirements of academic freedom and the teaching organisation's goals. On the one hand, a teaching organisation must have tools to monitor its procedures and to guide them towards its goals. On the other hand, researchers and teachers need freedom to choose how to implement high-quality research and teaching. Finding the balance between these two needs is one of a teaching organisation's challenges.

The interviews revealed several potentially problematic issues concerning the instructional process. First, the *goals in the formal documents were stated on an abstract level*, which made the work of the planning officer challenging. Second, the

[33] Many students at TKK need more than five years to receive the Master of Science degree. Actions have been planned to help more students to graduate in due time (five years).

goals and the planning of the studies were substance oriented. The goal setting and the planning do not start from the skills and knowledge that students should learn but from the course content. One practical consequence of this is that *soft skills are not discussed systematically during the planning process and thus they are not taught systematically either*. Third, the instructional process was steered by conventions and practical problems. Actions were taken to solve problems that required attention. Thus, *the process was reactive rather than proactive by nature*. Additionally, only small changes were made to the process if the need for them was perceived. In this respect, the process resembles the individual teacher's teaching process that was discussed earlier in chapter eight. Fourth, the collection of the feedback was inadequate. For example, the feedback was collected from the course level but not from the study module level. In addition, even though the interviews revealed several feedback sources, *the collection of the feedback was not systematic from all sources*. Fifth, the collected *feedback was not always utilised*. For example, the planning officer made a summary of the course feedback and sent it with the comments to the departments. However, there was no information whether this summary was actually read and whether it was used to adjust the following instructional processes. The same shortcoming concerning the inefficient use of feedback at TKK was noticed by the Finnish Higher Education Evaluation Council[34]. Finally, the interviews also highlighted that the university had *inadequate feedback channels for employees*. This was especially problematic since the university is soon going to merge with two other universities. There are committees that prepare this merger but the personnel outside the committees seem to be unaware of what the committees actually do. In addition, it is not clear how one could affect the committee's work that is going to affect future of all personnel and students. Figure 43 summarises some of the aspects that affect the instructional process at the organisation level.

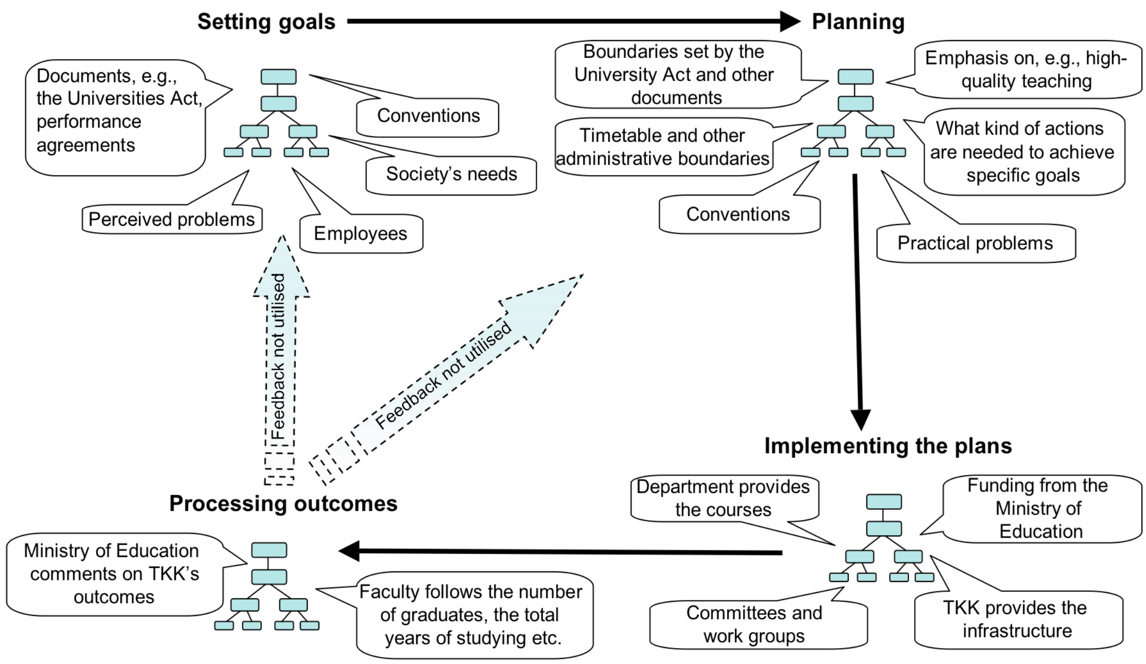


Figure 43 The instructional process from the organisation's point of view

[34] Teknillisen korkeakoulun laadunvarmistusjärjestelmän auditointi (Audition of TKK quality management system). Korkeakoulujen arviointineuvoston julkaisuja 1:2008. Tampere: Tammer-Paino Oy.

10 Discussion

The general goal of this research project was to shed light on the challenging and difficult aspects of the instructional process in computer science, especially in an introductory programming course (CS1). The research goal was divided into three main research questions, which focused on the instructional process from the students', teachers' and organisation's points of view, respectively. The research process resulted in both analysis models and empirical contributions.

Computing education research (CER) and computer science education research (CSER) at the university level are relatively young research areas. Many research contributions come from Anglo-American countries. The effect of Anglo-American research has resulted in a strong emphasis in the field on research that is related to educational psychology. For example, many studies focus on special topics of the instructional process. In this respect, this study provides a new perspective by using didactics as the basis for the research analysis. A holistic approach to the instructional process is visible throughout both the theoretical part and the empirical part of the study. The general system theory (GST) provided a useful theoretical framework to the instructional process in a context of computer programming courses, primarily at a technical university.

The contributions of this study are two-fold. On the one hand, this study presents analysis models that can be used to systematically categorise the research that is published in the field of CER and CSER, and analyse the instructional process. On the other hand, the empirical results highlighted the challenges and difficulties during the instructional process from students', teachers', and teaching organisation's points of view. This chapter discusses first the developed analysis models and then the empirical results. After that, the section 10.4 illustrates how one of the developed models, the feedback loop model, can be used to systematically analyse instructional processes.

10.1 Developed analysis models

In order to be able to tackle the general research goal of this study, three analysis models were created. First, the "*dimension doughnut*" was developed to discuss the multi-dimensional nature of the instructional process (see chapter four). The starting point for the development of this first analytical model was based on the work of Meisalo (1985). The "dimension doughnut" visualised nine dimensions (axis of coordinates), each representing a different point of view from which the instructional process could be analysed. The nine dimensions were organised into three groups: *actors, goals* and *means*. The group *actors* contained two dimensions: the students and the teaching organisation. These dimensions represented the actors of the instructional process. The second group *goals* included three dimensions that were identified as the content, the instructional entity and the level of cognitive process. These three dimensions together represented the aim of the instructional process. The third group *means* included four dimensions: the didactic activity, the level of concretising, the study activity and the interaction between students and teacher or students and students. These four dimensions represented the means by which the teachers and the students achieve the goals.

Each of the introduced dimensions is a large concept in it self. Therefore, each dimension was further divided into categories that emphasised the different levels of the dimension. For example, one of the dimensions was named "teaching organisation". However, teaching organisation has several levels. For instance, the Ministry of Education represents a very high organisational level. Universities with faculties and departments represent another. Furthermore, teachers at the university can also be perceived as a part of the teaching organisation as they implement on their part the goals and plans of the organisation. This study focused mainly on two dimensions of the "dimension doughnut", the student and the teaching organisation (see Figure 44). However, aspects of many other dimensions were also discussed during the research process.

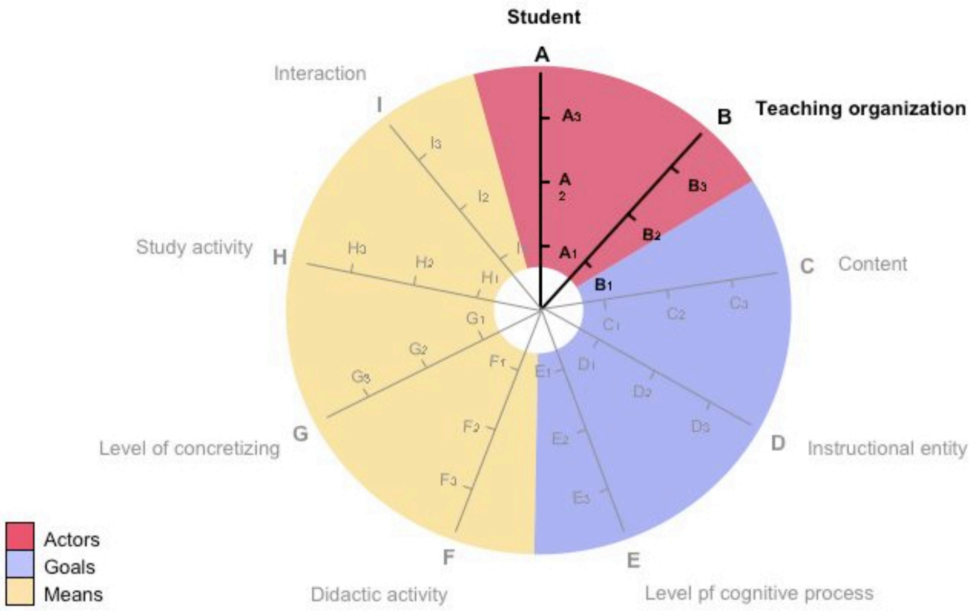


Figure 44 The dimensions that were analysed in this study

The *three-layered didactic triangle* was the second analysis model that was developed (Figure 45, see also chapter four). The original didactic triangle (see e.g., Peterssen 1989) and the didactic triangle developed by Kansanen (2003) were extended to three layers that allowed the discussion of instruction not only on a course level, but also on an organisation and society level. The three-layered triangle also provided a base for a categorisation system of the research that was based on didactic focus. Therefore, the new version of the didactic triangle provided a tool to discuss the instructional processes in larger contexts. Second, it provided a starting point for categorisation of research based on the didactic focus of this research, which was a novel approach in the field of computing education research.

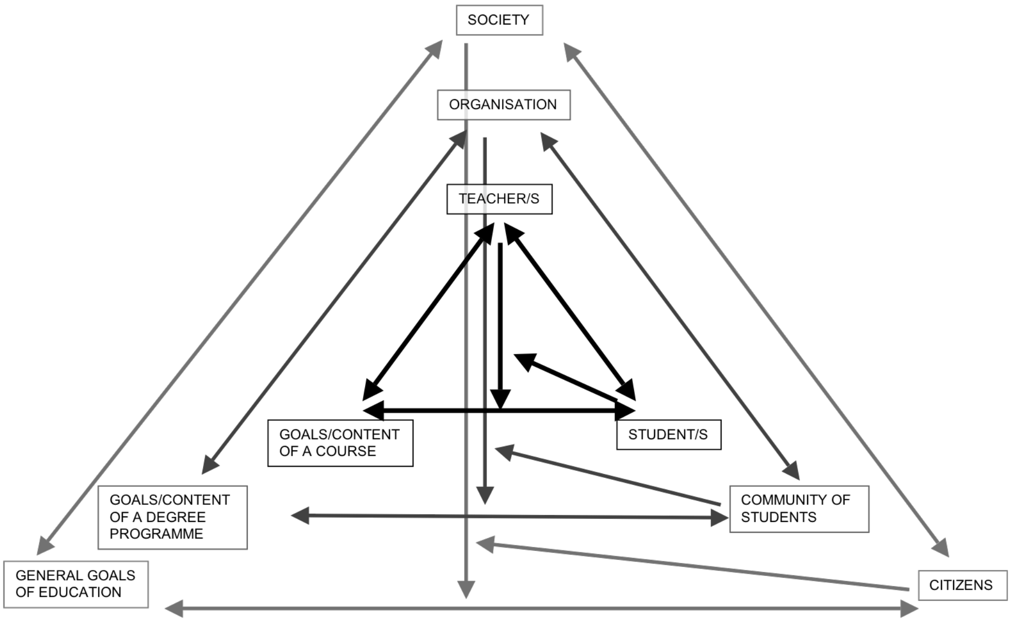


Figure 45 The three-layered didactic triangle (see section 4.3 for details)

The third analysis model was the *feedback loop model* of the instructional process (Figure 46, see also chapter five). The model was further developed from the existing models (see e.g. Meisalo 1985 and Meisalo et al. 2003) in accordance with the general system theory (GST). The modified model highlighted the role of feedback and guided the data collection and the analysis phases. The model provided a framework to analyse the instructional process systematically. For example, the model gave tools to analyse what kind of feedback was available and how it was utilised at different phases of the instructional process.

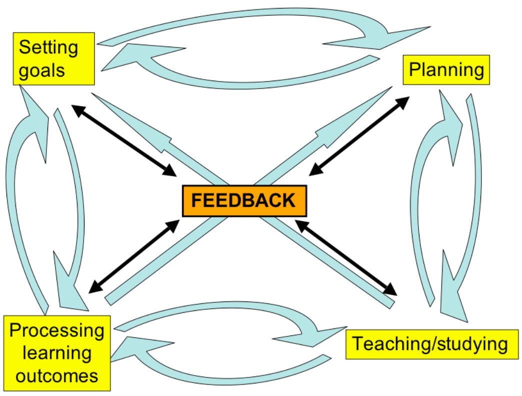


Figure 46 The feedback loop model (see section 5.2 for details)

10.2 Empirical results

The empirical results focused on three points of view of the instructional process: students, computer science teachers, and the teaching organisation. The students' point of view highlighted the difficulties students faced during an introductory programming course (CS1) and the reasons for dropping out of the course. The teachers' point of view focused, on the one hand, on teachers' conceptions of studying and their conceptions of how they can affect it and on the other hand, how teachers' perceived the teaching process. The teaching organisation's point of view highlighted first the educational quality related themes that were found in the formal documents. Second, this part of the study shed light on the conceptions of the representatives of the administration on the challenges of the instructional process and the functions of the university.

Students

The empirical results suggest that the reasons behind the students' decisions to drop out of the introductory programming course are manifold and that the reasons tend to cumulate. On average, the students who dropped out reported ten reasons that contributed to their decision. In addition, on average four reasons affected their decision critically. The first part of the student related research project revealed that students who had several reasons to drop out tended to do so earlier in the course than students who had fewer reasons to drop out. In a similar way, students who had friends in the same course dropped out later than the students who did not have friends in the same course. The second part of the student related research project highlighted the cumulative and concurrent nature of the reasons for dropping out of the course. This part of the study also shed light on the background of often mentioned reasons, such as the lack of time or motivation to study. Thus, the study provided information why student did not have enough time or motivation.

The empirical results suggested that reasons for dropping out of the introductory programming course could be placed into five categories (see the results of the factor analysis in chapter seven). *The course arrangements and not getting enough help* was one reason. In particular, students experienced that they did not get enough help from the assistant teachers or the lecturer. The closer analysis of the responses corroborated this result. Students who dropped out of the course found that asking for help from assistant teachers or friends was a less viable strategy in order to get over difficult issues during the course than did students who passed the course. *Difficulties in understanding course topics* was the second reason to drop out. Students had difficulties with understanding the issues that were covered in the course. They also had difficulties with the programming exercises. The third reason to drop out related to *time management and students' preferences.* The students had difficulties estimating how much time the course would require. Moreover, students had family and work related commitments that required time, too. The fourth reason for dropping out of the course related to the fact that *dropping out did not have consequences.* The course was either voluntary for the student or the course was not a prerequisite for other courses. Therefore, dropping out did not affect students' other studies. The fifth and last reason to drop out concerned students' study related preferences. Students *preferred other courses* to the introductory programming course and when they needed to choose a course to drop, they chose CS1.

The results concerning the difficulties students face during the instructional process in a CS1 course corroborate the difficulties found in the literature. For example, low motivation (Bergin and Reilly 2006; Forte and Guzdial 2005), time related aspects (Xenos et al. 2002) along with attitudes and abilities of the course personnel (Meisalo, Sutinen et al. 2002) have been found to cause difficulties in CS1 courses. The literature review also revealed several course content related aspects that students may find challenging to learn. The empirical results of this study are in line with the findings of the literature review. For example, the students' who dropped out of the CS1 course found designing of own code difficult. Literature review highlighted several studies that had identified designing in general to be difficult for students (Garner et al. 2005; Goldman et al. 2008; Lahtinen et al. 2005; Robins et al. 2003; Spohrer and Soloway 1986).

On the one hand, this study corroborates the earlier found difficulties. On the other hand, this study provides a comprehensive analysis of the reasons to drop out of a CS1 course. In addition, the analysis revealed the cumulative nature of the reasons. Furthermore, with the help of the instructional process model it was possible to place the found difficulties into different phases of the process. This systematic analysis provides information that can be used when the instructional processes are developed further.

Teachers

Computer science teachers' conceptions of studying divided into three main categories that focused on *knowledge (category A), action (category B),* and *student (category C)*. Each of these main categories divided further into sub-categories. The study by Bruce and Gerber (1995) provide a framework that can be used in comparing this study's results to the existing literature. However, the study by Bruce and Gerber and this study have a subtle difference in their focus. The study by Bruce and Gerber focused on teachers' conceptions of learning and how learning is achieved, whereas this study focuses on teachers' conceptions of studying[35]. The difference is that this study aims at obtaining a more comprehensive picture of the process of studying while stressing the comprehensive nature of the studying process. That is also the reason why I have chosen to use the word studying instead of learning in this study.

In order to relate results of this study to the existing knowledge of teachers' conceptions of studying and learning the computer science teachers' quotes were placed into the categories found in the study by Bruce and Gerber (1995) (Table 42). Even though this study and the study by Bruce and Gerber have subtle differences in their focus, it was possible to map the most of the quotes to the existing categories. Quotes in the categories that focused on knowledge & skills and actions mapped to Bruce and Gerber's categories, which concerned learning. Quotes in the categories that focused on the action mapped to Bruce and Gerber's categories, which focused on how learning is achieved. The distribution of the quotes suggests that studying as a concept contains both learning and the actions to achieve learning. Table 42 highlights also that the CS teachers' conceptions of studying did not show as great a variety as the conceptions of the teachers in the study by Bruce and Gerber. However, this study does not explain the reason for the lower amount of variation in the CS teachers' conceptions. Further studies are needed to shed light on that aspect.

---

[35] In Finnish opiskelu

Table 42 Teachers' conceptions of what learning and studying is

According to teachers' conceptions **learning is about** (Bruce and Gerber 1995) *According to teachers' conceptions **studying is** (this study)*	According teacher's conceptions **learning is achieved by** (Bruce and Gerber 1995) *According to teachers' conceptions **studying is** (this study)*
Acquiring knowledge *A.1 Piling up knowledge*	Application of study techniques, achieved by individuals *B.1 Hard work*
Absorbing and understanding *A.2 Acquiring knowledge and skills* *A.3 Goal-oriented acquiring of knowledge and skills of entire discipline*	Application of study techniques, being actively engaged in learning *B.2 Collecting knowledge* *B.3 Goal-oriented actions*
Development of thinking skills, ability to reason *C.2 Student's skill*	Individual effort or participation to social interaction (not necessarily 'observable' actions)
Developing the competences	Experiential process, experience with 'real life' problems
Changing attitudes, beliefs, behaviour *C.3 Goal-oriented (mental) process that aims at understanding, internalising, applying*	Experiencing alternative ideas etc. through discussion or experiencing contexts that are novel to the learner
Engaging in participative pedagogical experiences	Students in different situations learn differently

Note. The categories found in this study are printed in red in the table whereas Bruce and Gerber's categories are printed in black.

More detailed knowledge of the computer science teachers' conceptions was found when the focus of this study was turned on more definitive questions, such as, which aspects of studying are focal for successful studying and which aspects cause difficulties for students. Five main categories that related to the studying of computer science emerged: *student, knowledge, environment, problem solving* and *programming in a broader context*. The categories *student* and *knowledge* were further split into subcategories that included *student characteristics, student skills, theory and concepts,* and *ability to apply knowledge*. Each of the main categories was regarded as a focal prerequisite for successful studying. However, most of the mentions related to the categories *student characteristics* and *ability to apply knowledge*. When teachers were asked to elaborate on the their conceptions of which aspects were difficult for the student the subcategories *theory and concepts* and *ability to apply knowledge* were the most often mentioned categories.

When the computer science teachers' conceptions of the difficult aspects of studying were placed side by side with the students' reasons to drop out of the course, an interesting observation could be made. In Table 43, rows summarise the difficulties in three large categories (student, goal/content, and organising teaching and administrative procedures) that describe the main focus of the difficulty. The categories summarise the teachers', the students', and the representatives of administration's conceptions of difficult or problematic aspects during the instructional process. On the left side column are the aspects of studying that the CS teachers found difficult for the students. In the middle, there is the list of students' reasons for dropping out of the course. On the right side column, are the aspects of the instructional process that the representatives of the administration found problematic.

Table 43 Teachers', students', and the representatives of the administration's conceptions of problematic aspects of an instructional process

	Teachers' conceptions of aspects of studying that are difficult for the students	Students' reasons to drop out	Representatives of administration's conceptions of problematic aspects
STUDENT	• Students' skills: e.g. time management, ability to seek information • Problem solving skills • Student characteristics: e.g. ability to think logically, motivation	• Difficulties with the time management • Prefers other courses: e.g. a lot of other courses at the same time	
GOAL/ CONTENT	• Theory and concepts: e.g. level of abstraction • Ability to apply knowledge: e.g. reading and writing program, practising programming	• Difficulties in understanding course topics	• Goals abstract • Goals are not related to the planning and implementing of plans • Substance oriented planning
ORGANISING TEACHING AND ADMINISTRA-TIVE PROCEDURES		• Course arrangements and lack of help: e.g. course personnel's actions • Dropping out does not have consequences	• Feedback is not always utilised • Collection of feedback is not systematic • No feedback channels for personnel
ENVIRONMENT	• Studying environment		

Two aspects on the teachers' list can be found in the list of students' reasons to drop out. Many teachers thought that *theory and concepts,* and *applying the knowledge* are difficult for the students. The list of students' reasons to drop out of the course highlights *difficulties with understanding the course content* as one of the reasons. Likewise, the teachers thought that *time management* was difficult for students and it indeed was one of the reasons why students dropped out of the course. Interestingly, the teachers did not think that *course arrangements* were causing difficulties for the students; however, the *course arrangements* were one of the students' reasons to drop out. *Prefers other courses* was another reason to drop out that was not on the teachers' list. This reason to drop out may relate, on the one hand, to the students' motivation to learn programming and, on the other hand, to the pressure placed on the students by the demanding model study plan.

The CS teachers were also asked to elaborate on the possible ways they could affect the students' study process. Their responses indicated that, the planning and teaching phases in the teachers' process are in a central role. The decisions made during those phases affect students' setting goals, planning and studying phases. For example, during the planning phase, the teacher decides on the number and timetable of programming exercise rounds as well as on the ways students receive feedback during the course. These decisions should affect students' planning and studying phases, and on the type of feedback students can utilise during the course.

The results concerning the teachers' experiences of the teaching process highlighted the reality of how the instructional process proceeds. As a general remark, several actors collaborated during the instructional process. For example, several teachers sat down to discuss the content and goals of the courses. During the actual planning and teaching phases there were several assistant teachers involved in the process as well. The content of the course was in a central role when the goals of the process were set. Customs steered the planning and teaching phases in many respects. The courses were organised mainly the way they had been organised before and only small adjustments or changes were made when needed. Yet, in large-scale courses, the teachers were hesitant to make big changes since poorly implemented change would affect several hundreds of students. Teachers reported that there were several different feedback sources available. However, the student feedback form at the end of the course was the only source that was organised by the department. If the teachers wanted feedback from other sources (e.g., from assistant teachers), they had to collect it by themselves. Additionally, feedback from some sources did not reach the teacher. For example, the teaching organisation collected feedback from alumni students from time to time, but this feedback did not reach the teachers individually.

The utilisation of the feedback was not always straightforward. The teachers collected feedback during the course but it was not always possible to utilise it right a way. Often it was possible to make some changes to the next course but not during the ongoing course. In a large-scale course, programming exercises and lectures had to be planned many weeks ahead before the actual teaching phase. Therefore, even though the teacher observed that students would need some extra time to learn a certain topic she/he was not able to deviate from the original timetable. Moreover, the resources of the course set some limits to the teacher concerning what kind of teaching methods were applicable often due to the teacher-student ratio at the course. The literature review emphasised only some research concerning CS teachers. For example, the study by Sheard and Carbone (2007) corroborates the difficulties teachers face when the time table and course material has to be prepared well in advance. This study contributes to the field of computer science education by providing additional information on computer science teachers' conceptions on studying and teaching processes and the challenges teachers face.

Teaching organisation

The empirical results concerning the teaching organisation's point of view to the instructional process revealed how the instructional process was represented in formal documents, and how it was addressed by the representatives of the administration. The themes concerning *research based higher education* and *high quality research, education and teaching* were evident in the society, university, faculty and department level documents. The theme concerning *shorter study times* came up at the Helsinki University of Technology level and the faculty and department levels. Some new goals came up in the faculty and department level documents. For example, one of the goals was *a high-quality basic education*. The most concrete plans on how to achieve the goals and outcomes of the process were found in the faculty and department level documents. Faculties and departments were relatively free to organise their actions in the manner that they wanted. The teaching organisation gave only general resolutions that guided the faculties' and departments' actions.

The interviews of the representatives of the administration revealed how the teaching organisation level process is seen by them and what types of problematic aspects the process includes. As a general observation, the instructional process at the teaching organisation level was steered by conventions and timetables set by bureaucratic procedures. The analysis emphasised several aspects of the process that were perceived inadequate or problematic. First, the goal statements in the formal documents were at a very high level of abstraction. For the most part, these formal goals did not play a part during the actual planning and implementing of the plans. Second, the setting goals and planning phases were substance oriented. This resulted, for example, that *soft skills* were not systematically discussed or taught at the courses. Third, the collection of the feedback was not systematic or feedback was not collected at all. For example, feedback collection from the industry was incidental. In addition, the feedback was collected from the course level and from the degree level but not from the study module level. Fourth, the feedback was not always utilised. Finally, the interviews emphasised that there were not enough feedback channels for the personnel and therefore they could not give effective feedback and affect the university's future.

As Table 43 illustrates the computer science teachers, the students, and the representatives of the administration have some common conceptions of what causes challenges and difficulties during the instructional process. In Table 43 the challenges and difficulties were organised into four categories: *student, goal/content, organising teaching/administrative procedures,* and *environment*. In the *student* category, both the teachers and the students saw the student's inadequate time management skills as an aspect that caused difficulties. All three actors (teachers, students, and representatives of the administration) identified difficulties that related to the goals/content category. The difficulties that the teachers and the students identified focused on the course content. The difficulties that the representatives of the administration emphasised, related to the goals of the instructional process. Finally, both the students and the representatives of the administration indicated that the way of organising the courses and the administrative procedures were causing difficulties. These sources of difficulties relate to the prevailing culture at the university, i.e. accustomed ways of actions.

10.3 The interplay between students', teachers' and organisation's instructional processes

This research highlighted the instructional process from three different points of view. All points of view provided a different but complementary perspective to the rich reality in classrooms in a university of technology. The students, teachers and representatives of the administration represented different organisational levels that each had their own boundaries and realities that defined what they could do. However, these three levels or three points of view interrelate in many ways and affect each other (Figure 47).

In order to illustrate the interplay between the different points of view I take the studied introductory programming course (CS1) as an example case. A student enrols in an introductory programming course during his/hers second year of study at the university. His/hers decision is based on the study plan that the teaching organisation publishes. The study plan states that the course is a part of his/hers compulsory studies and the model timetable suggests that the student enrols in the course during his/hers second study year. The teaching organisation also provides student counselling in case the student needs help in planning studies. These are examples of how the teaching organisation steers the student's studying process (arrow 1). Some of the difficulties that the student experiences during the course may be due to the actions of the teaching organisation. For example, the model timetable might be too demanding and if the student enrols in all the courses listed in the timetable, he/she might run out of time. The student is forced to prioritise some courses and drop out of others.

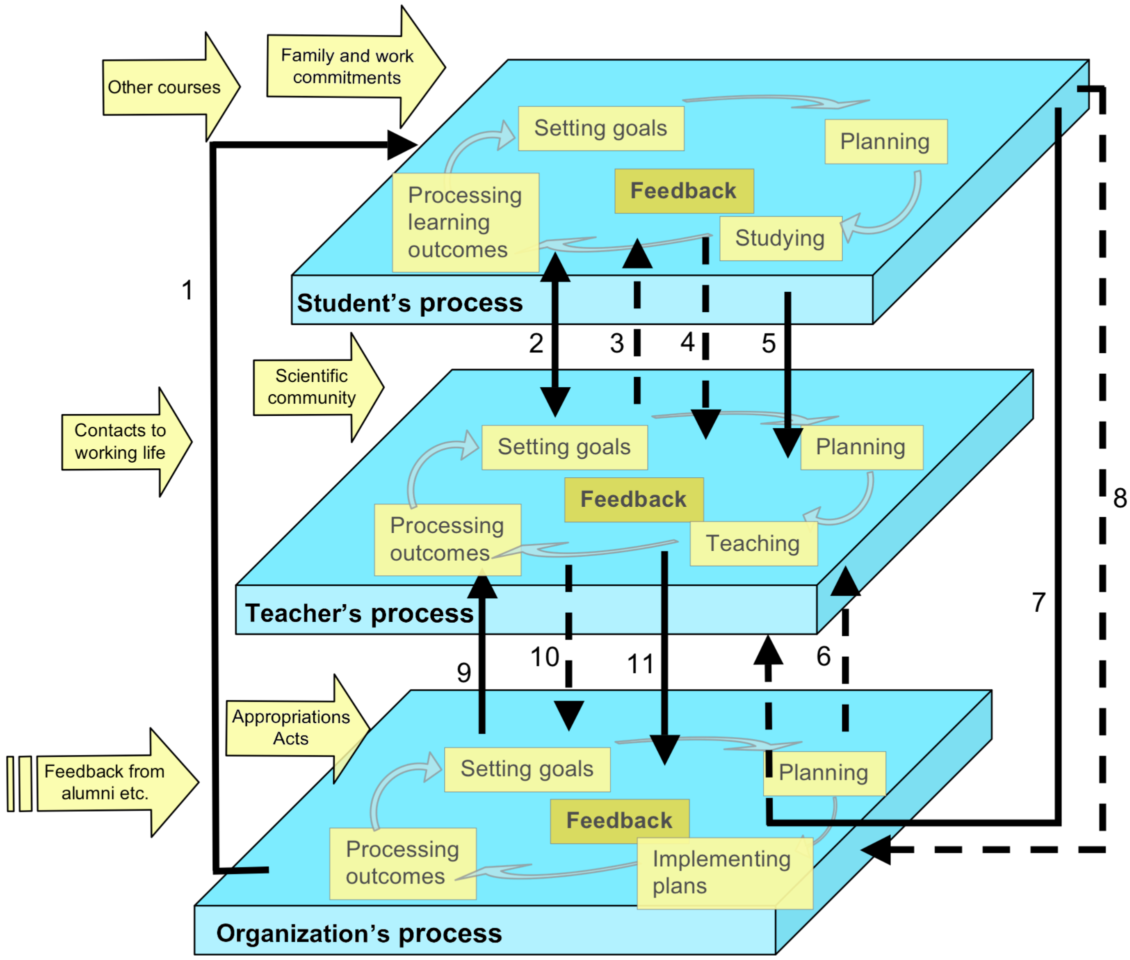


Figure 47 Interactions between the three levels of the instructional process

During the course, the student interacts with the teacher and the assistant teachers (arrow 2). The student attends lectures together with more than 500 other students. He or she also attends the programming exercise groups and reads the course news group on a regular basis. If the student has decided not to read the course book, this is an example of a pedagogical action that the teacher had planned but the student did not utilise (dashed line arrow 3). Based on the questions the student asks during lectures, in the exercise groups or in the news group, the teacher and the assistant teachers receive some information on what the student has learned and with which aspects he/she is struggling. However, since the introductory programming course has several hundreds of other students, the teacher and the assistant teachers are not able to follow an individual student's learning process closely. For instance, the teacher receives the results of the weekly exercises after the exercise round is closed. The teacher may notice that a student has serious difficulties with some of the salient concepts. However, the teacher is not able to respond to them (dashed line arrow 4). It is not possible, for example, to change the content of the following lectures and spend some more time to revise the topics that have already been taught. After the course, the student fills in the course feedback form, which the teacher reads (arrow 5).

The teaching organisation collects feedback from different sources. For instance, it asks the graduates to fill out a questionnaire concerning their studies as a whole. These surveys might reveal also something about individual course and how meaningful graduates perceived them to be. The organisation also receives feedback from working life. However, this feedback does not reach the teachers (dashed line arrow 6). The teaching organisation receives some information on the students' studying processes and the courses by systematically collecting the course feedback from all courses (arrow 7). The planning officer makes a summary of the feedback and gives it back to the departments. Teachers in the department could then use the summarised feedback to develop teaching. Based on the data it appears that departments do not take this summarised feedback into account (arrow 7). This does not imply that individual teachers would not read and utilise the course feedback in their work. However, the data did imply that the feedback is not systematically utilised in a larger context. The department did not give guidelines nor did it monitor how the organisation collected and utilised the feedback. The interviews revealed some feedback sources that the organisation did not utilise at the time of the interviews. The teaching organisation also did not systematically collect information on how students studies proceeded (dashed line arrow 8).

The teaching organisation provides teachers with resources to organise the course (arrow 9). The organisation also sets some general goals for the education. However, individual teachers are not always aware of these goals (arrow 6). The data did not reveal systematic measures that the organisation would use to get feedback from the teachers' processes. In fact, one of the deficiencies that unfolded was that the university did not have enough feedback forums for its employees (dashed line arrow 10). On the other hand, the organisation received some information on how the teachers' and students' processes were proceeding. At least the organisation was informed if there were some serious difficulties in some particular course (arrow 11).

The description of the case highlighted some examples of how the different levels may interact and how the missing feedback loops can be identified. In addition to interrelating levels, there are external factors that affect the instructional processes. For example, a student's studying process is affected by other concurrent studies that also require the student's attention and time. Family and work commitments may also require the student's time. There are external factors that affect the teacher's process, as well. For instance, the teacher has his or her own collaborators in the industry and the scientific community. These contacts may influence, for instance, the teacher's opinion on what he/she should teach and how. The teaching organisation receives appropriation and general guidelines from the society that affect the teaching organisation's functions. The organisation also collects feedback from different sources, such as alumni and trade unions. However, this collection is not systematic, which means that feedback is at times collected arbitrarily.

10.4 The feedback loop model in use: analysis of the challenges of the student, teacher and the representatives of organisation

In this section, I will use case examples to illustrate how the feedback loop model can be used to systematically analyse instructional processes. The case examples highlight, for example, during which phases of the process feedback is missing and how does it affect the rest of the process.

The first case example is the instructional process of a second year communications engineering student who took the introductory programming course as a part of his compulsory studies, but dropped out of the course. The student had some programming experience before entering the course and he was interested in programming. His goal was to get a passing grade for the course and learn more about programming. He also planned to take advanced programming courses later. The student planned to get 38 study weeks during the spring semester. According to his self-estimation, he usually tended to underestimate the time he needed to invest in a course. According to the interview data, the student did not receive feedback concerning his goals or plans (nor did he look for feedback by himself during those phases) (dashed line arrows 1 and 3 in Figure 48). He also did not refine his goals after the planning (dashed line arrow 4 in Figure 48). During the studying phase, the student attended only few lectures and spent most of the time studying independently. During the studying phase, he received feedback from the automatic grading system; the feedback seemed to be adequate in guiding his studies (arrow 6 in Figure 48). At the time when the student submitted his project plan for the course, problems started to emerge. On the one hand, the student perceived that the feedback the assistant teacher gave concerning the project plan was not constructive and that the grading of the plan was not fair. The student received a low grade for the project plan (arrow 10 in Figure 48). On the other hand, the student had many exams and projects, which he had to do for his other courses. In addition, he already knew he had some time in the following summer to take the programming course again. These factors resulted in the student dropping out of the course.

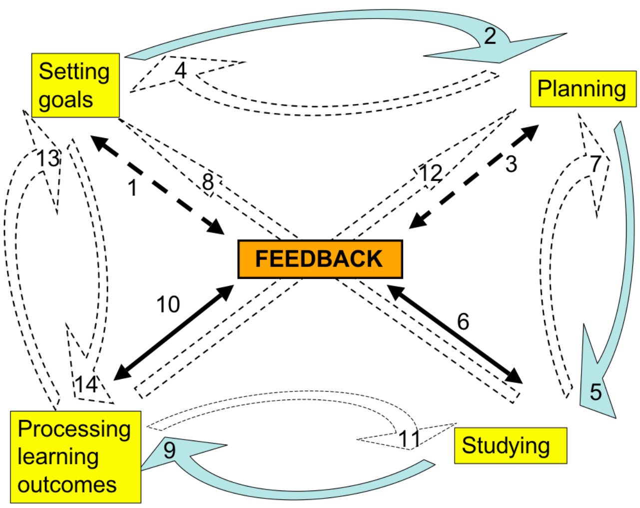


Figure 48 The instructional process of one student who dropped out of the course

The student's case, as analysed through the feedback loop model (Figure 48), draws attention to the many missing arrows in the model. The student did not receive feedback during the setting goals nor the planning phases. During the studying phase, the student got feedback but it did not give him any reason to adjust his goals, plans or studying. During the processing learning outcomes phase, the student perceived that the feedback was not constructive and therefore the feedback did not lead to him taking any corrective measures.

In general, the feedback loop model was not able to capture all the possible reasons why the students dropped out of the course. For example, the model is not sensitive to students' personal problems at home, which may have distracted the student's concentration for the course. However, the feedback loop model is capable of highlighting course related aspects that the teacher can affect.

The next case example is an introductory programming course teacher's instructional process. The case describes the instructional process during one course as it was seen by one of the interviewed teachers. In the beginning of the instructional process, no goals were actively set; the goals of the course were not considered every year. Rather, the goals were inherited from the previous years' courses; the goals stayed the same or almost the same for several years. The teacher received the feedback concerning the goals during the faculty discussions that concerned the course's goals in relation to other course's goals. In addition, the teacher used his own background and experiences to reflect on the goals (arrow 1 in Figure 49). During the planning phase, the teacher reflected and revised the plans according to his previous years' experiences (arrow 3 in Figure 49). However, there was no corrective loop to revise the goals since they stayed nearly static for several years. During the teaching phase, the teacher gave lectures and collected feedback from different sources. The teacher read the course news group, monitored the time students spent doing programming exercises and collecting feedback after lectures (arrow 6 in Figure 49). The planning and the teaching phases overlapped partly. The teacher did some refinement to the lecture plans based on the feedback he got during the teaching phase (arrow 7 in Figure 49). The goals of the course were not reviewed during or after the teaching phase (dashed line arrow 8 in Figure 49). At the processing learning outcomes phase, the teacher was able to see the results of the programming exercises and written exam. In addition, the statistics on average grades and the time students spent served as a feedback source for the teacher (arrow 10 in Figure 49). However, since the learning outcomes and feedback came with delay and not right after the teaching phase there were no corrective loops to goals, plans or teaching (arrows 13, 12, 11 in Figure 49).

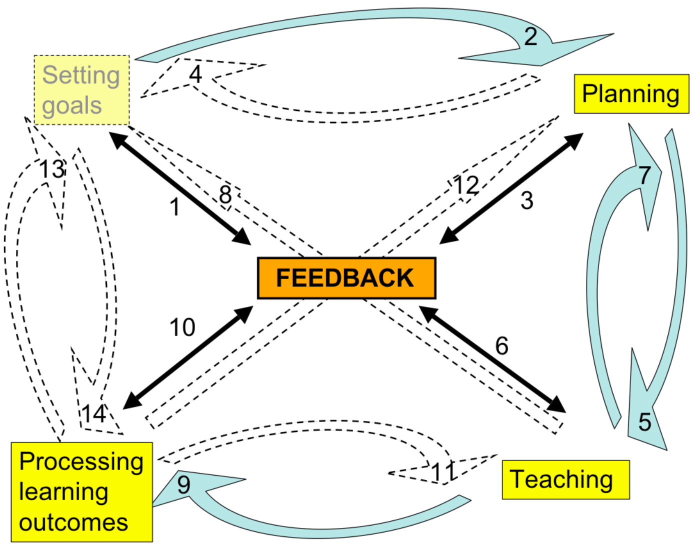


Figure 49 The teacher's process

As on the student's feedback loop, there are several missing arrows in the teacher's feedback loop model. Many arrows are missing because the teacher is not able to do instant adjustments in a large-scale course. Instead, in many cases, the feedback the teacher receives affects the next course he is giving. Figure 49 portrays the instructional process during one course and thus many of the arrows do not exist in the figure. Furthermore, the figure may look different when it is applied to smaller-scale courses.

The last case example is the planning officer's instructional process. She had several administrative tasks going on at the same time. This case example focuses on the teaching organisation's ambition concerning the shorter completion times of studies. The concrete goal was that students, on average, would get their Master's degree in five years. This goal was stated in the performance agreement between the university and the Ministry of Education and the Faculty inherited that goal from the higher administrative level. Therefore, there was no goal setting phase since the Faculty was given the goal of helping students graduate on time. The planning officer was a part of the team that surveyed the statistics and discovered that only few students graduate in five years (arrow 14 in Figure 50). Therefore, the goal was perceived relevant by the planning officer and other administrative staff. The planning officer received feedback concerning the goal from the higher organisation levels that had set the goal in the first place (arrow 1 in Figure 50). The planning phase contained plans to provide students better support. In addition, there were also plans concerning the ways to find out why it takes more than five years to graduate. The data did not highlight any specific feedback during planning phase nor did it indicate any revisions to the goals. Therefore, the lines three and four in Figure 50 are missing.

During the implementing plans phase, the student advisors offered support to the students. They also contacted students whose studies were not proceeding as expected to find out what the problems were and whether the university could do something about them. The data did not highlight any feedback sources that the planning officer would have used during the implementing plans phase (arrow 6 in Figure 50). During the processing outcomes phase the planning officer and other administrative staff expected to see the statistics concerning how many years students have been studying before graduating. Comparison of statistics and the earlier set goals provided the feedback (arrow 10 in Figure 50). However, this particular instructional process is an ongoing process. The teaching organisation has just started to implement the plans and the outcomes of those measures can only be seen after a few years. Therefore, arrows 11 and 12 in Figure 50 are somewhat hypothetical at this point. Based on the outcomes and the information the organisation collects the organisation may think about the sizing of the courses (arrow 12 in Figure 50) or decide to revise the model study plan to make it more realistic (arrow 11 in Figure 50).

These three case examples illustrate how the feedback loop model can be used to systematically analyse instructional processes. For instance, the model helped to identify the phase of the instructional model that was related to the student's reasons for dropping out of the course. The missing arrows in the models can be used as starting points for further discussions on why feedback loops are not there and whether it would be beneficial for the students, teachers or administrative staff to have those loops.

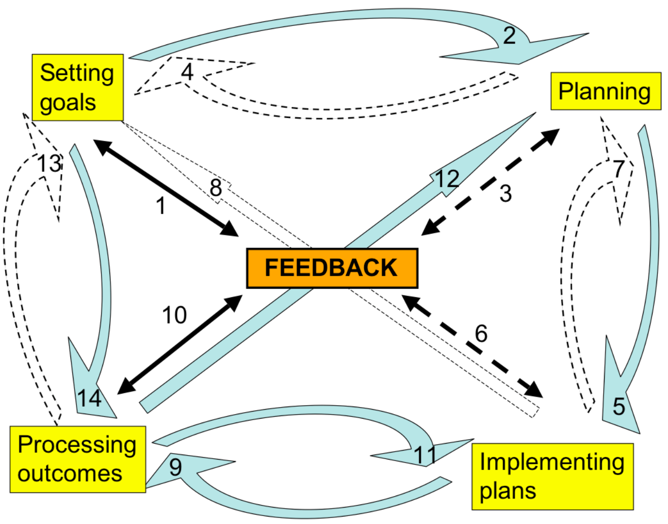


Figure 50 The teaching organisation's process

10.5 To summarise

In the introduction, I stated that I am trying to shed light into the black box called the instructional system of an introductory programming course (CS1). At first, the box seemed like a system that received inputs (e.g., students and resources) and from the other end came the output (e.g., the passing rate of a course). Yet, it was not well known why the system produced such outputs. Different aspects of the instructional process have been studied before. However, this study used a holistic approach for understanding the instructional process. The results of this research highlighted several challenging and interrelating aspects of the instructional processes that would benefit from well-designed interventions that touch all three levels of the instructional process (Figure 47). The analysis models and empirical contributions of this research may be used as guidelines when planning such interventions.

I chose a holistic approach to better grasp the challenging and difficult aspects of the instructional process. The decision was based on the observation that focusing on one point of view would result in only partial information that would not be very useful when planning concrete interventions. The instructional processes do not happen without external influences and the different level processes interplay with each other. Students' have their own challenges and so do the teachers and the teaching organisation. If I had focused on only one point of view I would not had understood the richness of the different factors that lead to, for example, the high drop-out rate in an introductory programming course.

The development of education is a challenging undertaking that requires time and resources. In order for the development work to be systematic and effective, it needs both background knowledge and tools. This thesis has provided both; the empirical results provided the background knowledge on the challenges the students, teachers and teaching organisation face during the instructional processes. Meanwhile the analysis models provided tools that can be used to analyse the educational research and the different levels of instructional processes systematically. The empirical results of this thesis verify that these tools have been able to shed light on some focal challenges in the instructional process. Thus, this study provides abstract tools that can be used when practical interventions are planned, implemented, and evaluated.

The restriction of the holistic approach was that I had to stay at a general level. I was not able to solely focus on some particular point of view and gather detailed and comprehensive information. Rather, I had to take a step backward to see the big picture. This resulted the disappearance of the details. However, I strongly believe that gaining the holistic understanding was worthwhile.

The strength of the analysis models that were developed for this work lies in their level of abstraction. The three analysis models that I developed can be used to analyse teaching and studying in many contexts. They are not restricted to some particular subject matter or the age level of the students. The empirical data concerned mostly an introductory programming course. The data highlighted that some of the found difficulties related directly to the content of the course. However, the study revealed also generic difficulties that are transferable to other contexts. More importantly, when all results were reviewed side by side, it became evident that the students', teachers' and teaching organisation's processes interrelate. This observation indicates that interventions to lower the drop-out rate need to address all three levels in order to be effective.

11 Conclusions

This study has provided a holistic picture of the instructional process by looking at it from three different viewpoints. The contributions of this thesis include three analysis models (“dimension doughnut”, three-layered didactic triangle, feedback loop model) and empirical results. Analysis models helped to identify several sub-areas and topics within computer science education where more research is needed, and to analyse the instructional process systematically. The empirical results have highlighted the challenges students, teachers and representatives of the administration confront during the instructional process. The results suggest that the development of education is a challenging task. The reasons behind the students’ decisions to drop out of the course were manifold and they tended to cumulate individually. The teachers of large-scale courses received feedback from students, but they were not always able to utilise it during the course. At the teaching organisation level the collection of feedback was not systematic or in some cases feedback was not collected at all. The results suggest that there is a need to improve the instructional process at each of the three levels. For instance, in order to lower the course’s drop-out rate, interventions are needed at the course level and at the organisational level. In the following paragraphs, some of the results are highlighted and may be used as guidelines for the university.

One of the reasons for dropping out of the introductory programming course was the course arrangements. Especially, students experienced that they did not get enough help.

Results highlighted that students who dropped out of the introductory programming course (CS1) found asking for help from teaching personnel and friends a less viable strategy to get over a difficult issues during the course than did the students who passed the course. The answers to the open ended questions and the results of the interview also suggested that not all assistant teachers were able to explain course topics the way student would have understood them. One possibility to reduce this sort of reason for dropping out is to provide pedagogical training for teachers and assistant teachers. The training provides the course personnel with the knowledge and skills to develop the course so that it better supports students’ studying. Particularly, the pedagogical training of the assistant teachers is essential, since students interact with them frequently when they do the programming assignments. Programming is the part of the course where students may encounter concrete challenges; for instance, they have difficulties in debugging their program. The assistant teachers’ ability to interact with the students and to scaffold the learning is important for how students perceive the situation. The pedagogical training of the assistant teachers is also important because they often are undergraduate students with no previous teaching experience nor pedagogical training.

Since one of the teaching organisation’s goals is to provide high quality teaching, it also is the organisation’s duty to provide the pedagogical training to enhance the teachers’ and assistant teachers’ skills. Therefore, the difficulties that relate to the course arrangements do not only apply to individual teachers, but also to the teaching organisation. However, in order to be able to understand the course level difficulties, the teaching organisation has to have a systematic way of collecting and utilising the course feedback.

One of the reasons for dropping out of the introductory programming course was that students had difficulties with time management.

Time managing and planning for studies are essential parts of the students' studying skills. Both the course teacher and the teaching organisation can help the students with their time managing difficulties. One solution is that the teacher clearly states the amount of time the course is likely to take. Weekly submitted small assignments for the course may also help the students to study a little every week instead of leaving everything to the last minute. However, these measures do not help all students. The teachers of the CS1 courses at TKK have implemented the previously mentioned actions for years with a little success. Anecdotal evidence, that is based on informal discussions with course teachers, suggest that some students in the course have to spend substantially more time to complete the programming exercises than an average student in the course. This provides motivation for future research on students' difficulties concerning specific course topics.

The teaching organisation can help the students with their time managing difficulties by providing enough student advisors who can support and enhance the students' study skills. The demanding model study plan might also relate to the time management problems. If students try to take several demanding and laborious courses during the same semester there is a high risk that those students will run out of time. Therefore, the teaching organisation could revise the model study plan so that it guides the students to plan schedules that are more realistic.

Another reason for dropping out of the introductory programming course was the students' lack of motivation.

The reasons behind poor motivation related to the students' general study motivation and the experienced difficulty of the course. Some dropped out students also thought that the payoff between the amount of work the course required and an outcome of the course were imbalanced. For instance, some students felt that the number of received study credits was too low compared the time they had to invest to the course. Another student felt that it was not fair that it took so many hours to code a simple program that counted the mean of three numbers. The teacher of the course may help the students by showing them that programming is a relevant skill; moreover, examples that relate to students' future studies or duties might help increase the students' motivation and curiosity. Additionally, the teacher could view the course content from the students' point of view and see how it is most beneficial for the students' needs.

The teaching organisation is responsible of the coherence and the content of degree programs. The introductory programming course is included in many degree program curricula since it is regarded as a basic skill that engineers should possess. However, for the student it might be difficult to see the relevance of the basic studies to his or hers major studies or future career. The teaching organisation could explain the relevance of basic studies, for instance, in the study guidebook.

At the teachers' and the organisation's levels the setting goals and planning phases were often content oriented.

The strong focus on the content at the *setting goals* and *planning* phases resulted in that skills and more generic goals of courses and broader study entities were not systematically taken into account. For instance, according to the interview study the soft skills were not taught systematically during the studies. The teaching organisation and teachers should consider ways of widening their focus from content to broader skills that the students should posses once they graduate or finish a study entity. The teaching organisation could, for instance, organise committees or other meetings for the teachers where they would discuss the skills a typical student needs to possess after finishing a study entity, such as a minor studies in computer science[36]. After agreement on the basic skills these skills could then be divided into course level goals. This procedure would help the organisation and individual teachers to compose a coherent curriculum that teaches students the skills they need after graduation.

At the teaching organisation level the collection of feedback was not always systematic or feedback was not collected at all. The utilisation of the collected feedback was also not systematic.

The systematic collection and analysis of feedback should be made a part of the teaching organisations' yearly procedures. The organisation should develop systematic procedures to collect feedback from different sources; such as the course, the major and minor studies, and the degree level studies. On the other hand, the sources should include different perspectives from students, teachers, and administrative staff to representatives of the alumni, employers, and the scientific community.

For example, the teaching organisation could send out surveys to employers and recently graduated alumni every two or three years to receive the feedback on how the organisation has managed to educate professionals to meet the needs of the society. At the same time, the organisation needs to develop a systematic procedure to utilise the feedback. When the feedback has been received, the organisation should make sure that also individual teachers receive feedback.

Future work

This study has provided analysis models for systematic studying the instructional process of an introductory programming course (CS1). However, the models may be used in studying other courses as well. This study has also provided three points of view to the challenges of the instructional process. Further research is needed to fully understand what happens during the different phases of the instructional process and the challenging aspects of the process. For instance, the teacher's planning phase includes several actions that were not analysed in detail in this study. The teacher is not free to plan the course the way he/she wants but instead there are rules, customs, and restrictions that guide the planning phase. A closer analysis of the processes in setting goals, planning, teaching and processing learning outcomes is needed to fully understand which aspects of the processes cause difficulties and what could be done to help teachers to overcome the difficulties. This kind of analysis is closely related to the development of the teaching organisation's processes.

The interplay between students, teachers, and the teaching organisation provides a starting point for future research. For instance, it would be interesting to analyse the way students, teachers, and the representatives of the administration perceive each others and how they see the interplay between the three actors. For instance, how teachers' perceive students and the teaching organisation's actions, how students perceive teachers and the teaching organisation, and how representatives of administration perceive teachers and students.

[36] Just recently, in the spring 2009, a committee was formed to consider the skills students need to possess after finishing a larger study entity.

Another interesting future study topic can also be created based on the current situation. The teaching organisations are under constantly changing pressures, especially at Helsinki University of Technology, Helsinki School of Economics and University of Art and Design Helsinki. These three universities are just about to merge into one university. This merge is no doubt challenging for all parties. However, at the same time, it is a great opportunity to create something new. The merging of the three universities provides an interesting starting point for further studies. One option would be, for instance, to analyse the way the three universities collect and utilise feedback now. The results of the analysis, the deficiencies and well working practices, could then be used as starting points for the development of new practices for the new university.

The development of education is an ongoing task. Even in cases where the basic structure of the education is working and no alarming sings are calling for immediate interventions, there is always a need for proactive development and improvement. The expectations society places to universities are high and the universities have to adapt to constantly changing requirements. Therefore, universities must keep on developing their education in order to be able to provide the skills and fundamental knowledge base that society expects. Research on teaching, studying and learning is needed so that the development work can be based on systematically collected and analysed knowledge rather than gut feeling and anecdotal evidence.

References

(2004). "Regional strategy for education and research up to 2013", D. f. E. a. S. Policy, (ed.). Ministry of Education.

(2008). "Development plan for education and research 2007- 2012 (Kouluts ja tutkimus 2007-2012. Kehittämissuunnitelma)". Ministry of Education.

ACM. (2001). *Computing Curricula 2001 Computer Science*.

ACM. (2005). *Computing Curricula 2005*. ACM and IEEE.

AERA. (2007). "12 divisions within AERA". Retrieved October 24, 2007 from http://www.aera.net/divisions/Default.aspx?menu_id=62&id=179

Andersson, R., and Bendix, L. (2005) "Towards a Set of eXtreme Teaching Practices." *Presented at Proceedings of the 5th Baltic Sea conference on Computing education research (Koli Calling 2005)*, Koli National Park, Finland.

ASEE. (2007). "6th Annual ASEE Global Colloquium on Engineering Education. Program Overview." Retrieved November 26, 2007 from http://www.asee.org/conferences/international/2007/Program.cfm

Bai, G., and Lindberg, L.-Å. (1999). "A sociocybernetic approach to information systems development." *Kybernetes*, 28(6/7), 792-809.

Baillie, C., and Fitzgerald, G. (2000). "Motivation and attrition in engineering students." *European Journal of Engineering Education*, 25, 2.

Bandura, A. (1997). *Self-efficacy: the exercise of control*, New York: Freeman.

Barker, L. J., and Garvin-Doxas, K. (2004). "Making Visible the Behaviors that Influence Learning Environment: A Qualitative Exploration of Computer Science Classrooms." *Computer Science Education*, 14(2), 119-145.

Beaubouef, T., and Mason, J. (2005). "Why the high attrition rate for computer science students: some thoughts and observations." *SIGCSE Bull.*, 37(2), 103-106.

Ben-Ari, M. (2001). "Constructivism in computer science education." *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45-73.

Ben-Ari, M. (2004). "Situated Learning in Computer Science Education." *Computer Science Education*, 14(2), 85-100.

Ben-Ari, M. (2005). "Situated learning in this high-technology world." *Science & Education*, 14(3-4).

Ben-Ari, M., and Yeshno, T. (2006). "Conceptual models of software artifacts." *Interacting with Computers*, 18, 1336-1350.

Bennedsen, J., and Caspersen, M., E. . (2005). "An investigation of potential success factors for an introductory model-driven programming course" *Proceedings of the first international workshop on Computing education research*. Seattle, WA, USA.

Bennedsen, J., and Schulte, C. (2008) "What does “Objects-First” Mean?— an International Study of Teacher Perception of Objects-First." *Presented at The 7th Baltic Sea Conference On Computing Education Research*, Joensuu.

Bennett, R. (2003). "Determinants of undergraduate student drop out rates in a university business studies department." *Journal of Further and Higher Education*, 27(2), 123-141.

Bergamin, P. (2006). *Blended Learning: Die wiedergefundene Gemainschaft. Konzeptionelle Betrachtungen zur Umsetzung von Lernszenarien im Fernstudium auf Hochschlstufe.*, IFeL - Institut für Fernstudien- and eLearningforscung.

Bergin, S., and Reilly, R. (2006). "Predicting Introductory Programming Performance: A multi-institutional multivariate study." *Computer Science Education*, 16(4), 303-323.

Bergin, S., Reilly, R., and Traynor, D. (2005). "Examining the role of self-regulated learning on introductory programming performance" *Proceedings of the first international workshop on Computing education research*. Seattle, WA, USA.

Berglund, A. (2005). *Learning computer systems in a distributed project course. The what, how and where*, Uppsala: Uppsala Universitet.

Berglund, A. (2006). "Phenomenography as a way to research learning in computing." *Bulletin of the National Advisory Committee on Computing Qualifications, BACIT*, 4(1).

Berglund, A., and Eckerdal, A. (2006). "What do CS students try to learn? insights from a distributed, project-based course in computer systems." *Computer Science Education*, 16(3), 185-195.

Bertalanffy, L. von. (1972). "The history and Status of General System Theory." *The Academy of Management Journal*, 15(4), 407-426.

Bertrand, Y., and Houssaye, J. (1999). "Pédagogie and didactique: An intcestuous relationship." *Instructional Science*, 27, 33-51.

Birnbaum, R. (1988). *How Colleges Work. The Cybernetics of Academic Organisation and Leadership*, San Francisco: Jossey-Bass Publisher.

Booth, S. (1992). *Learning to program. A phenomenographic perspective*, Gothenburg, Sweden: Acta Universitatis Gothoburgensis.

Boyle, R., Carter, J., and Clark, M. (2002). "What Makes Them Succeed? Entry, progression and graduation in Computer Science." *Journal of Further and Higher Education*, 26(1), 3-18.

Bruce, C., and Gerber, R. (1995). "Towards university lectures' conceptions of student learning." *Higher Education*, 29(4), 443-458.

Byrne, P., and Lyons, G. (2001). "The effect of student attributes on success in programming" *Proceedings of the 6th annual conference on Innovation and technology in computer science education*. Canterbury, United Kingdom.

Cantwell Wilson, B. (2002). "A Study of Factors promoting Success in Computer Science Including Gender Difference." *Computer Science Education*, 12(1-2), 141-164.

Cantwell Wilson, B., and Shrock, S. (2001). "Contributing to success in an introductory computer science course: a study of twelve factors" *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*. Charlotte, North Carolina, United States.

Carbone, A. (2007). *Principles for Designing Programming Tasks: How task characteristics influence student learning of programming*, Monash University.

Carbone, A., Mannila, L., and Fitzgerald, S. (2007). "Computer science and IT teachers' conceptions of successful and unsuccessful teaching: A phenomenographic study." *Computer Science Education*, 17(4), 275-299.

Chamillard, A. T. (2006). "Using student performance predictions in a computer science curriculum" *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*. ACM: Bologna, Italy.

Chen, D., and Stroup, W. (1993). "General System Theory: Toward a Conceptual Framework for Science and Technology Education for All." *Journal of Science Education and Technology*, 2(3), 447-459.

Chowdhury, S., Endres, M., and Lanis, T. W. (2002). "Preparing Students For Success In Team Work Environments: The Importance Of Building Confidence." *Journal of Managerial Issues*, 14(3), 346.

Clark, B., R. (1979). "The many pathways of academic coordination." *Higher Education*, 8(3), 251-267.

Cohen, L., Manion, L., and Morrison, K. (2000). *Research Methods in Education*, London: Routledge Farmer.

Cook, A., and Leckey, J. (1999). "Do Expectations Meet Reality? A survey of changes in first-year student opinion." *Journal of Further and Higher Education*, 23(2), 157-171.

Cope, C., and Ward, P. (2002). "Integrating learning technology into classrooms: The importance of teachers' perceptions." *Educational Technology & Society*, 5(1), 67-74.

Covington, M. V. (2000). "Goal Theory, Motivation, and School Achievement: An Integrative Review." *Annu. Rev. Psychol*, 51, 171-200.

Crenshaw, T., L. , Chambers, E. W., and Metcalf, H. (2008). "A case study of retention practices at the University of Illinois at Urbana-Champaign." *SIGCSE Bull.*, 40(1), 412-416.

Dale, N., B. (2006). "Most difficult topics in CS1: results of an online survey of educators." *SIGCSE Bull.*, 38(2), 49-53.

de Raadt, M., Toleman, M., and Watson, R. (2007) "Incorporating programming strategies explicitly into curricula." *Presented at Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007)*, Koli National Park, Finland.

Dick, W., Carey, L., and Carey, J. O. (2004). *Systematic Design of Instruction*: Allyn & Bacon.

du Boulay, J. B. H. (1986). "Some difficulties of learning to program." *Journal of Educational Computing Research*, 2(1), 57-73.

Eckerdal, A. (2006). *Novice Students' Learning of Object-Oriented Programming*, IT Licentiate thesis, Uppsala University, Uppsala.

Eckerdal, A. (2009). *Novice Programming Students' Learning of Concepts and Practise*.

Edwards, J., R. (1992). "A Cybernetic Theory of Stress, Coping and Well-Being in Organisations." *The Academy of Management Journal*, 17(2), 238-274.

EERA. (2006). "The list of strands in ECER-2006 ". Retrieved on November 12, 2007 from https://bei.leeds.ac.uk/dbsql/AFreeQueryServlet?*id=1&*formId=index&*context=BEIC&Conference=ECER-2006&*servletURI=https://bei.leeds.ac.uk/dbsql/AFreeQueryServlet

EERA. (2007). "The list of strands in ECER-2007 ". Retrieved on November 12, 2007 from https://bei.leeds.ac.uk/dbsql/AFreeQueryServlet?*id=1&*formId=index&*context=BEIC&Conference=ECER-2007&*servletURI=https://bei.leeds.ac.uk/dbsql/AFreeQueryServlet

Elliot, A. J., and Church, M. A. (1997). "A Hierarchical Model of Approach and Avoidance Achievement Motivation." *Journal of Personality and Social Psychology*, 72(1), 218-232.

Engeström, Y. (1987). *Learning by expanding: An activity-theoretical approach to developmental research.*, Helsinki: Orienta- Konsultit.

Engeström, Y. (1990). *Learning, Working and Imagining. Twelve Studies in Activity Theory*, Helsinki: Orienta-Konsultit Oy.

Engeström, Y. (1999). "Expansive Visibilization of Work: An Activity-Theoretical Perspective." *Computer Supported Cooperative Work*, 8(1-2), 63-93.

Fazey, D., and Fazey, J. (1998). "Perspectives on motivation: The implications for effective learning in higher education", in S. Brown, S. Armstrong, and G. Thompson, (eds.), *Motivating students*. London: SEDA: Staff and Educational Development Series, pp. 59-72.

FERA. (2003). "The annual Finnish conference in educational research. The description of paper session topics. (in Finnish)". ". Retrieved on October 24, 2007 from http://www.kasvatus.net/kasvatustieteenpaivat2003/ryhmat.html

FERA. (2004). "The annual Finnish conference in educational research. The description of paper session topics. (in Finnish)". Retrieved on October 24, 2007 from http://www.kasvatus.net/kasvatustieteenpaivat2004/teemaryhmat.html

FERA. (2005). "The annual Finnish conference in educational research. The description of paper session topics. (in Finnish)". Retrieved on October 24, 2007 from http://www.kasvatus.net/kasvatustieteenpaivat2005/index.html

FERA. (2006). "The annual Finnish conference in educational research. The description of paper session topics. (in Finnish)". Retrieved on October 24, 2007 from http://www.kasvatus.net/kasvatustieteenpaivat/teemaryhmat/teemaryhmat.html

FERA. (2007). "The annual Finnish conference in educational research. The description of paper session topics. (in Finnish)". Retrieved on October 24, 2007 from http://www.kasvatus.net/kasvatustieteenpaivat2007/teemaryhmat/teemaryhmat.html

FERA. (2008). "The annual Finnish conference in educational research. The description of paper session topics. (in Finnish)". Retrieved on December 29, 2008 from http://congress.utu.fi/ktpaivat/teemaryhmat.php

Fincher, S., and Petre, M. (2004). *Computer Science Education Research*, London: Taylor & Francis.

Foor, C., E., Walden, S., E., and Trytten, D., A. (2007). "“I Wish that I Belonged More in this Whole Engineering Group:” Achieving Individual Diversity." *Journal of Engineering Education*, 96 (2), 103-115.

Forte, A., and Guzdial, M. (2005). "Motivation and Nonmajors in Computer Science: Identifying Discrete Audiences for Introductory Courses." *IEEE Transactions on Education*, 48(2), 248-253.

Friesen, N. (2006) "Microlearning and (Micro)Didaktik." *Presented at Micromedia & e-Learning 2.0: Gaining the Big Picture. Proceedings of Microlearning Conference*, Innsbruck.

Gal-Ezer, J. (1995). "Computer Science Teachers' Certification Program." *Computers Educ.*, 25(3), 163-168.

Garner, S., Haden, P., and Robins, A. (2005). "My program is correct but it doesn't run: a preliminary investigation of novice programmers' problems" *Proceedings of the 7th Australasian conference on Computing education - Volume 42*. Australian Computer Society, Inc.: Newcastle, New South Wales, Australia.

Garvin-Doxas, K., and Barker, L. J. (2004). "Communication in computer science classrooms: Understanding defensive climates as a means of creating supportive behaviors." *ACM Journal of Educational Resources in Computing*, 4(1), 1-18.

Glaser, B., G., and Strauss, A., L. (1967). *The discovery of grounded theory: strategies for qualitative research*, Chicago: Aldine.

Glass, R., L. , Ramesh, V., and Vessey, I. (2004). "An analysis of research in computing disciplines." *Commun. ACM*, 47(6), 89-94.

Goldman, K., Gross, P., Heeren, C., Herman, G., Kaczmarczyk, L., Loui, M., C., and Zilles, C. (2008). "Identifying important and difficult concepts in introductory computing courses using a delphi process: selective compression of unicode arrays in java" *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. Portland, OR, USA.

Goold, A., and Rimmer, R. (2000). "Factors affecting performance in first-year computing." *SIGCSE Bull.*, 32(2), 39-43.

Gruba, P., Moffat, A., Søndergaard, H., and Zobel, J. (2004). "What drives curriculum change? ." *Presented at Proceedings of the sixth conference on Australasian computing education.* Dunedin, New Zealand.

Haberman, B., Lev, E., and Langley, D. (2003). "Action research as a tool for promoting teacher awareness of students' conceptual understanding." *SIGCSE Bull.*, 35(3), 144-148.

Haho, P. (unpublished manuscript). *The role of individual and organizational learning in process innovations*, Espoo, Finland: SimLab, TKK.

Hattie, J., and Timperlay, H. (2007). "The Power of Feedback." *Review of Educational Research*, 77(1), 81-112.

Heinonen, V. (1989). *Kasvatustieteen perusteet (Fundamendals of Education)*, Jyväskylä: Gummerus Kirjapaino Oy.

Herrmann, N., Popyack, J., L. , Char, B., Zoski, P., Cera, C., D. , Lass, R., N. , and Nanjappa, A. (2003). "Redesigning introductory computer programming using multi-level online modules for a mixed audience" *Proceedings of the 34th SIGCSE technical symposium on Computer science education*. Reno, Navada, USA.

Hostetler, K. (2005). "What Is "Good" Education Research?" *Educational Researcher*, 34(6), 16-21.

Johnson, B., and Christensen, L. (2004). "Educational Research. Quantitative, qualitative, and mixed approaches." 33(7), 14-26.

Jungert, T. (2008). "A longitudinal study of engineering students' approaches to their studies." *higher Education Research & Development*, 27(3), 201-214.

Kansanen, P. (1999). "Teaching as Teaching-Studying-Learning Interaction." *Scandinavian Journal of Educational Research*, 43(1), 81-89.

Kansanen, P. (2002). "Didactics and its Realation to Educational Psychology: Problems in Translating a Key Concepts Across Research Communities." *International Review of Education*, 48(6), 427-441.

Kansanen, P. (2003). "Studying-the Realistic Bridge Between Instruction and Learning. An Attempt to a Conceptual Whole of the Teaching-Studying-Learning Process." *Educational Studies*, 29(2/3), 221-232.

Kansanen, P., and Meri, M. (1999). "The didactic relation in the teaching-studying-learning process." *TNTEE Publications*, 2(1), 107-116.

Katz, S., Allbritton, D., Aronis, J., Wilson, C., and Soffa, M. L. (2006). "Gender, achievement, and persistence in an undergraduate computer science program." *SIGMIS Database*, 37(4), 42-57.

Kember, D., and Kwan, K.-P. (2000). "Lecturers' approaches to teaching and their relationship to conceptions of good teaching." *Instructional Science*, 28, 469-490.

Kinnunen, P., and Malmi, L. (2005). "Problems in Problem-Based Learning - Experiences, Analysis and Lessons Learned on an Introductory Programming Course." *Informatics in Education*, 4(2), 193-214.

Kinnunen, P., and Malmi, L. (2006). "Why students drop out CS1 course?" *Proceedings of the 2006 international workshop on Computing education research*. Canterbury, United Kingdom.

Kinnunen, P., and Malmi, L. (2008). "CS minors in a CS1 course" *Proceeding of the fourth international workshop on Computing education research*. Sydney, Australia.

Kinnunen, P., McCartney, R., Murphy, L., and Thomas, L. (2007). "Through the eyes of instructors: a phenomenographic investigation of student success" *Proceedings of the third international workshop on Computing education research*. Atlanta, Georgia, USA.

Kolikant, Y. B-D.., and Pollack, S. (2004). "Community-oriented pedagogy for in-service CS teacher training." *SIGCSE Bull.*, 36(3), 191-195.

Künzli, R. (2000). "German Didaktik: Models of Re-presentation, of Intercourse, and of Experience", in I. Westbury, S. Hopmann, and K. Riquarts, (eds.), *Teaching as a reflective practice: the German Didaktik tradition*. Mahwah, NJ: Lawrence Erlbaum Associates.

Kvale, S. (1996). *Interviews. An Introduction to Qualitative Research Interviewing*, London: SAGE Publications.

Lahdes, E. (1997). *Peruskoulun uusi didaktiikka* (in Finish), Helsinki: Otava.

Lahtinen, E., Ala-Mutka, K., and Järvinen, H.-M. (2005). "A study of the difficulties of novice programmers." *SIGCSE Bull.*, 37(3), 14-18.

Lahtinen, E., and Ahoniemi, T. (2005). Visualizations to Support Programming on Different Levels of Cognitive Development. *Proceedings of the Koli Calling 2005 Conference on Computer Science Education*. Koli National Park, Finland.

Lattu, M. (2002). "The Nature of the Finnish Technology Education." *Presented at Technology Ecucation in Practice. Seminar on Technology Education at the University of Oulu / Faculty of Education, University of Luleå / "House of Technology" and Central Ostrobotnia Polytechnic / Ylivieska Campus*.

Lincoln, Y., and Guba, E. (1985). *Naturalistic Inquiry*, Newbury Park: Sage Publications.

Lindblom-Ylänne, S., Trigwell, K., Nevgi, A., and Ashwin, P. (2006 June). "How approaches to teaching are affected by discipline and teaching context." *Studies in Higher Education*, 31(3), 285-298.

Lister, R., Adams, E., S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Mostr, J. E., Sanders, K., Seppälä, O., Simon, B., and Thomas, L. (2004). "A multi-national study of reading and tracing skills in novice programmers." *SIGCSE Bull.*, 36(4), 119-150.

Luhmann, N. (2002). *Das Erziehungssystem der Gesellschaft*, Frankfurt am Main: Suhrkamp.

Machanick, P. (2007). "A Social Construction Approach to Computer Science Education." *Computer Science Education*, 17(1), 1-20.

Magee, R., Baldwin, A., Newstead, S., and Fullerton, H. (1998). "Age, Gender and Course Differences in Approaches to Studying in First-year Undergraduate Students", in S. Brown, S. Armstrong, and G. Thompson, (eds.), *Motivating Students*. London and New York: Routledge Falmer. Taylor & Francis Group.

Mancy, R., and Reid, N. (2004) "Aspects of Cognitive Style and Programming." *Presented at 16th Workshop of the Psychology of Programming Interest Group*, Carlow, Ireland.

Martin, E., Prosser, M., Trigwell, K., Ramsden, P., and Bejamin, J. (2000). "What university teachers teach and how they teach it." *Instructional Science*, 28, 387-412.

Mayring, P. (2000). "Qualitative Content Analysis." *FQS Forum: Qualitative Social Research*, 1(2), ART. 20.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., Laxer, C., Thomas, L., Utting, I., and Wilusz, T. (2001). "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students." *SIGCSE Bull.*, 33(4), 125-180.

McDonald, M., and McDonald, G. (1999). "Computer science curriculum assessment." *ACM SIGCSE Bulletin*, 31(1), 194 - 197.

McKinney, D., and Denton, L., F. . (2004). "Houston, we have a problem: there's a leak in the CS1 affective oxygen tank" *Proceedings of the 35th SIGCSE technical symposium on Computer science education*. Norfolk, Virginia, USA.

Meisalo, V. (1985). "Futurology and education of physics and chemistry teachers." *Presented at Proceedings of the Nordic Conference on Science and technology Education: The Challenge of the Future*, Copenhagen.

Meisalo, V., Suhonen, J., Torvinen, S., and Sutinen, E. (2002). "Formative evaluation scheme for a web-based course design" *Proceedings of the 7th annual conference on Innovation and technology in computer science education*. Aarhus, Denmark.

Meisalo, V., Sutinen, E., and Tarhio, J. (2003). *Modernit oppimisympäristöt (Modern learning environments)*, Pieksämäki: RT-Print Oy.

Meisalo, V., Sutinen, E., and Torvinen, S. (2002). "How to improve a virtual programming course?" *The 32nd ASEE/IEEE Frontiers in Education Conference, FIE*. Boston, MA, USA.

Milne, I., and Rowe, G. (2002). "Difficulties in Learning and Teaching Programming - Views of Students and Tutors." *Education and Information Technologies*, 7(1), 55-66.

Namazi, K. H., and McClintic, M. (2003). "Computer Use Among Elderly Persons in Long-term Care Facilities." *Educational Gerontology*, 29(6), 535.

Niemivirta, M. (2004). *Habits of Mind and Academic Endeavors. The Correlates and Consequences of Achievement Goal Orientations*, Dissertation, University of Helsinki, Helsinki.

Niss, M. (1999). "Aspects of the Nature and State of Research in Mathematics Education." *Educational Studies in Mathematics*, 40(1), 1–24.

Onwuegbuzie, A. J., and Daniel, L. G. (2003). "Typology of analytical and interpretational errors in quantitative and qualitative educational research." *Current Issues in Education [On-line]*, 6(2). Retrieved May 2, 2009 from http://cie.asu.edu/volume6/number2/

Pears, A., Seideman, S., Eney, C., Kinnunen, P., and Malmi, L. (2005). "Constructing a Core Literature for Computing Education Research." *ACM SIGCSE Bulletin*, 37(4), 152-161.

Peterssen, W. H. (1989). *Lehrbuch Allgemeine Didaktik*, Munchen: Ehrenwirth.

Pillay, N., and Jugoo, V., R. (2005). "An investigation into student characteristics affecting novice programming performance." *SIGCSE Bull.*, 37(4), 107-110.

Pioro, B. T. (2006). "Introductory computer programming: gender, major, discrete mathematics, and calculus." *J. Comput. Small Coll.*, 21(5), 123-129.

Prosser, M., Martin, E., Trigwell, K., Ramsden, P., and Lueckenhausen, G. (2005). "Academics' experiences of understanding of their subject matter and the relationship of this to their experiences of teaching and learning." *Instructional Science*, 33, 137-157.

Prosser, M., and Trigwell, K. (1999). "Relational perspectives on higher education teaching and learning in the sciences." *Studies in Science Education*(33), 31-60.

Quinn, P. M. (2002). *Qualitative Research & Evaluation Methods*, Thousand Oaks (Calif.): Sage.

Ramalingam, V., LaBelle, D., and Wiedenbeck, S. (2004). "Self-efficacy and mental models in learning to program" *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*. Leeds, United Kingdom.

Randolph, J., Julnes, G., Lehman, S., and Sutinen, E. (2008). "A methodological review of computer science education research." *Journal of Information Technology Education*, 7, 135-162.

Rantanen, E., and Liski, E. (2009). *Valmiiksi tavoiteajassa. Teknillisen alan opiskelijoiden opintojen eteneminen ja opiskelukokemukset tekniikan kandidaatin tutkinnossa*, Espoo: Teknillinen korkeakoulu.

Reed, D., Miller, C., and Braught, G. (2000). "Empirical investigation throughout the CS curriculum" *Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*. Austin, Texas, United States.

Reges, S. (2006). "Back to basics in CS1 and CS2" *Proceedings of the 37th SIGCSE technical symposium on Computer science education*. Houston, Texas, USA.

Robins, A., Haden, P., and Garner, S. (2006). "Problem distributions in a CS1 course" *Proceedings of the 8th Austalian conference on Computing education - Volume 52*. Australian Computer Society, Inc.: Hobart, Australia.

Robins, A., Rountree, J., and Rountree, N. (2003). "Learning and Teaching Programming: A Review and Discussion." *Computer Science Education*, 13(2), 137-172.

Rountree, N., Rountree, J., Robins, A., and Hannah, R. (2004). "Interacting factors that predict success and failure in a CS1 course"*Working group reports from ITiCSE on Innovation and technology in computer science education*. Leeds, United Kingdom.

Ryan, G., and Bernard , R. (2003). "Data Management and Analysis Methods", in N. Denzin and Y. Lincoln, (eds.), *Collecting and Interpreting Qualitative Materials*. London: Sage Publication.

Samuelowicz, K., and Bain, J. (2001). "Revisiting academics' beliefs about teaching and learning." *Higher Education*, 41, 299-325.

Schulte, C., and Bennedsen, J. (2006). "What do teachers teach in introductory programming?" *Proceedings of the 2006 international workshop on Computing education research*. Canterbury, United Kingdom.

Schwartzman, L. (2006). "A qualitative analysis of reflective and defensive student responses in a software engineering and design course" *Proceedings of the 6th Baltic Sea conference on Computing education research (Koli Calling 2006)*. Uppsala, Sweden.

Schwartzman, L. (2007) "Student transformative learning in software engineering and design: discontinuity (pre)serves meaning." *Presented at Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007)*, Koli National Park, Finland.

Seel, H. (1999). "Didaktik as the professional science of teachers." *TNTEE Publications*, 2(1), 85-93.

Seppälä, O., Malmi, L., and Korhonen, A. (2005). "Observations on student errors in algorithm simulation exercises." *Presented at Proceedings of the 5th Baltic Sea conference on Computing education research (Koli Calling 2005)*, Koli National Park, Finland.

Sheard, J., and Carbone, A. (2007). "ICT teaching and learning in a new educational paradigm: lecturers' perceptions versus students' experiences." *Presented at Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007)*, Koli National Park, Finland.

Sheard, J., Simon, Hamilton, M., and Lönnberg, J. (2009). "Analysis of research into the teaching and learning of programming." *Proceedings of the fifth international workshop on Computing education research workshop*. Berkeley, CA, USA.

Simon. (2007). "A Classification of Recent Australasian Computing Education Publications." *Computer Science Education*, 17(3), 155-169.

Simon, Carbone, A., Raadt, M. d., Lister, R., Hamilton, M., and Sheard, J. (2008). "Classifying computing education papers: process and results" *Proceeding of the fourth international workshop on Computing education research*. Sydney, Australia.

Simon, Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., Raadt, M. d., Haden, P., Hamer, J., Hamilton, M., Lister, R., Petre, M., Sutton, K., Tolhurst, D., and Tutty, J. (2006). "Predictors of success in a first programming course." *Proceedings of the 8th Austalian conference on Computing education - Volume 52*. Hobart, Australia.

Simon, B., Hanks, B., Murphy, L., Fitzgerald, S., McCauley, R., Thomas, L., and Zander, C. (2008). "Saying isn't necessarily believing: influencing self-theories in computing" *Proceeding of the fourth international workshop on Computing education research*. Sydney, Australia.

Sorva, J. (2007) "Students' understandings of storing objects." *Presented at Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007)*, Koli National Park, Finland.

Sorva, J. (2008) "The same but different students' understandings of primitive and object variables." *Presented at Eighth Baltic Sea Conference on Computing Education Research (Koli Calling 2008)*. Koli National Park, Finland.

Spohrer, J., C. , and Soloway, E. (1986). "Novice mistakes: are the folk wisdoms correct?" *Commun. ACM*, 29(7), 624-632.

Stasko, J., Guzdial, M., Clancy, M., Dale, N., B., and Fincher, S. (2001). "Models and areas for CS education research." *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*. Charlotte, North Carolina, United States.

Stein, L. A. (1999). "Challenging the Computational Metaphor: Implications for How We Think." *Cybernetics and Systems*, 30(6), 473 - 507.

Stephenson, C., and West, T. (1998). "Language Choice and Key Concepts in Introductory Computer Science Courses." *Journal of Research on Computing in Education*, 31(1), 89.

Strauss, A., and Corbin, J. (1990). *Basics of Qualitative Research*, Newbury Park: SAGE Publications.

Surakka, S. (2005). *Needs Assessment of Software Systems Graduates*, Espoo, Finland: Otamedia Oy.

Tashakkori, A., and Teddle, C. (2003). *Handbook of Mixed Methods in Social & Behavioral Research*, London: SAGE Publications.

Tenenberg, J., Fincher, S., Blaha, K., Bouvier, D., Chen, T.-Y., Chinn, D., Cooper, S., Eckerdal, A., Johnson, H., McCartney, R., Monge, A., J-E, M., Petre, M., Powers, K., Ratcliffe, M., Robins, A., Sanders, D., Schwartzman, L., Simon, B., Stoker, C., Tew, A. E., and VanDeGrift, T. (2005). "Students designing software: a multi-institutional study." *Informatics in Education*, 4(1), 143-162.

Tight, M. (2008). "Higher education research as tribe, territory and/or community: a co-citation analysis." *Higher Education*, 55(5), 593-605.

Tinto, V. (1975). "Dropout from Higher Education: A Theoretical Synthesis of Recent Research." *Review of Educational Research* 45(1 (Winter, 1975)), 89-125.

Trigwell, K., and Prosser, M. (1991). "Improving the quality of student learning: the influence of learning context and student approaches to learning on learning outcomes." *Higher Education*, 22, 251-266.

Trigwell, K., Prosser, M., and Waterhouse, F. (1999). "Relations between teachers' approaches to teaching and students' approaches to learning." *Higher Education*, 37(1), 57-70.

Tucker, A., B. , Kelemen, C., F. , and Bruce, K., B. . (2001). "Our curriculum has become math-phobic!" *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*. Charlotte, North Carolina, United States.

Tutty, J., Sheard, J., and Avram, C. (2008). "Teaching in the current higher education environment: perceptions of IT academics." *Computer Science Education*, 18(3), 171-185.

Uljens, M. (2001). "On General Education as a Discipline." *Studies in Philosophy and Education*, 20, 291-301.

Vagianou, E. (2006). "Program working storage: a beginner's model" *Proceedings of the 6th Baltic Sea conference on Computing education research (Koli Calling 2006)*. Koli National Park, Finland.

Valentine, D. (2004). "CS educational research: a meta-analysis of SIGCSE technical symposium proceedings." *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, ACM SIGCSE Bulletin*, 35(1), 255-259.

Valentine, J., DuBois, D., and Cooper, H. (2004). "The Relation Between Self-Beliefs and Academic Achievement: A Meta-Analytic Review." *Educational Psychologist*, 39(2), 111-133.

Ventura, P. R. (2005). "Identifying Predictors of Success for an Object-First CS1." *Computer Science Education*, 15(3), 223-243.

Waite, W., M. , Jackson, M., H. , Diwan, A., and Leonardi, P., M. (2004). "Student culture vs group work in computer science" *Proceedings of the 35th SIGCSE technical symposium on Computer science education*. Norfolk, Virginia, USA.

Wenger, E. (1998). *Communities of Practice. Learning, Meaning, and Identity*, Cambridge: Cambridge University Press.

Wiedenbeck, S. (2005) "Factors Affecting the Success of Non-Majors in Learning to Program." *Presented at ICER ′05*, Seattle, Washington, USA.

Wood, R., and Bandura, A. (1989). "Social Cognitive Theory of Organizational Management." *Academy of Management Review*, 14(3), 361-384.

Xenos, M., Pierrakeas, C., and Pintelas, P. (2002). "A survey on student dropout rates and dropout causes concerning the students in the Course of Informatics of the Hellenic Open University." *Computers & Education*, 39, 361–194.

Xinogalos, S., Satratzemi, M., and Dagdilelis, V. (2006). "An introduction to object-oriented programming with a didactic microworld: objectKarel." *Computers & Education*, 47(2), 148-171.

Appendix

Appendix 1 Summary of arrows and phases of the instructional process

This table summarises the instructional process from the students' teachers' and organisations' point of view. It is a simplified description in a sense that it does not take into account all the possible factors but aims at highlighting the main phases of the process through examples.

	Student	Teacher	Organisation
Time scale for the process cycle	Days - weeks - months/ the duration of one course	Months to a year	Year(s)
Setting goals	A student sets goals for his/hers studies. The goals can concern a single course or a larger set of courses.	A teacher sets goals for the course. The curriculum states the general goals, which the teacher then divides on a more concrete level that are attainable within one course, and are in line with the goals of the whole degree.	The different level organisations set goals for themselves. The Universities Act defines the general mission to the University. For example, the Universities' mission is to provide higher education and strive for high international level of research, education, and teaching. Further, each University has its own missions and strategies that steer the goal setting at several organisational levels. The Council's goal is to provide requisites for the University to function according to its mission. The Council of the Helsinki University of Technology sets both quantitative and qualitative goals. The quantitative goals concern the number of degrees the university is going to produce next year. On the other hand, the Council states in the Degree Regulations the objectives for Bachelor's and Master's degrees. The Faculty sets goals concerning the number of the degrees it is going to produce next year. At the same time, the Faculty's general goal is to promote high-quality higher education. The Department sets quantitative goals concerning the

			number of the degrees it is going to produce next year. On the other hand, department sets goals concerning the content of the courses. It aims at developing the content of the degree programme as well as the content of single courses to appeal to the students and to provide them with the skills and knowledge that are needed after graduation.
Arrow 1	The student receives feedback at the setting goals phase and may adjust the goals accordingly. For example, the student may compare his/hers previous experiences of similar courses with the goals and adjust them as a result of self-reflection. Discussions with peers also help the student to evaluate the goals. Finally, the teacher of the course may help the student to consider the attainability of the goals.	The teacher receives feedback at setting goals phase and adjusts the goals accordingly. For example, the teacher can use his/her previous experiences from the same course to reflect on the current goals. Co-worker teachers may also provide feedback by sharing their experiences and expressing their opinion.	The different level of organisations get feedback at goals – phase. The feedback comes in a form of negotiations. The rector/Council may use his/hers own observations concerning the previous instructional processes as a source of feedback. On the other hand, the rector, as a representative of the Council of the Helsinki University of Technology, negotiates with the Ministry of Education. Negotiations concern the number of degrees the university agrees to produce in exchange for the certain amount of appropriation. During the performance negotiations, the rector receives feedback from the Ministry of Education. For example, the poor outcomes of the previous year may result cutbacks in following years' appropriation. The longer-term trends, for example, concerning the number of graduates serve a feedback for the Council and the Ministry of Education, too, and may affect the quantity of appropriation. Faculty negotiate with the Council (rector) about the number of degrees it agrees to produce in exchange for certain amount of appropriation and receives feedback during those negotiations. The longer-term trends concerning the number of graduates give feedback to the Faculty, too. On the other hand, the Faculty has content related goals that may change according to the feedback that industry provides. Faculty can decide, for instance, to change the emphasis on a degree structure if there emerges a need for it. The experiences from previous years instructional processes serve as a feedback source. The Department develops the goals and the content of the courses according to the emerging needs of the industry. It receives content related feedback by following the trends of

			the field. The previous years outcome (e.g., the number of degrees) serve as a feedback and steering mechanism for setting quantitative goals.
Arrow 2	The student transitions from the setting goals phase to the planning phase.	The teacher transitions from the setting goal phase to the planning phase.	The different level of organisation transfers to make plans how the goals could be met.
Planning	The student makes plans how he/she could attain set goals. For example, after setting goals to seek for certain grade the student makes plans how to meet those goals. The student estimates how much time the course will require and makes a schedule for him/herself. He/she can also make, for instance, more specific plans how he/she will go about studying.	The teacher makes concrete plans how the goals could be met. For instance, the teacher plans lectures, decides the content and number of the exercises. More specific plans may contain issues, such as how much time he/she will use to introduce new concepts, what kind of tools he/she will use to help the teaching (e.g., visualisation tools for algorithms).	Different levels of organisation make plans how the goals could be attained. For example, the rector negotiates with the Ministry of Education and Faculties concerning the number of the degrees that will be produced during the next year and distributes appropriations accordingly. The rector and the Council of the Helsinki University of Technology make plans about how much money will be distributed between teaching and research, and on what grounds this decision is based on. In addition, the Council and rector make plans what kind of additional services that support teaching are needed and how they are financed. The Council commits itself to the planning of the extent and subject matters of study entities that comprise basic studies. Faculty negotiates with the Departments concerning the number of degrees and study credits they are willing to produce in exchange of certain amount of appropriation. The faculty plans the basic study entities that are same and compulsory to all students at the Faculty. The Faculty also makes plans how the entity of the degree programmes would serve the needs of the industry and the field best. The faculty makes strategic plans concerning the emphasis of the research fields. This aims at on one hand on keeping up the standards the industry requires and on the other hand to attract good students that would graduate in time. On a practical level the Faculty co-ordinates the making of the lecture timetables and exams. Department makes plans how the appropriations are best distributed to meet the goals that it has. The plans are also made to assure that the supply of the courses are appropriate

			and meet the standards. In a longer time scale the Department also makes plans concerning the content of the degree programme and single courses.
Arrow 3	The student receives feedback at the planning phase. There are at least three sources of feedback: self-reflection, peers, and the teacher. For example, the student can reflect plans to his/hers self-efficacy beliefs and previous experiences with similar courses. How did I do last time when I was studying something similar? How did my plans work at that time? In addition, the student can reflect his/hers plans to his/hers peers' experiences. The course teacher and the assistant teachers can also provide feedback concerning the plans.	The teacher receives feedback at the planning phase. There are at least two sources of feedback available: self-reflection and other teachers. For example, the teacher can reflect his/hers previous experiences as a teacher in general or especially as a teacher of the same course. Contemplating on what degree did the plans proved to be viable earlier is a source of feedback. Another source of feedback comes from other teachers. The discussions with colleagues can provide valuable feedback concerning the plans. For example, how did other teachers organise the same or similar course? What kind of difficulties did they encounter and how did they handle them. What are their experiences of the same student population, what worked with them earlier and what did not work.	The different level of organisations get feedback at planning –phase. Setting goals and planning occur largely during the performance negotiations. Therefore, also the great deal of feedback is offered during these negotiations. However, it is sometimes difficult to distinguish whether the feedback concerns the setting goals or planning phase. For example, the Council of the Helsinki University of Technology receives the feedback during the performance negotiations with the Ministry of the Education and Departments concerning the distribution of appropriations. The previous instructional processes serve as a feedback source, too, during the planning. The Faculty receives feedback from the Council and the Departments during the negotiations that concern the distribution of appropriations. Setting goals and planning are closely tied and corrective loop may happen frequently during negotiations. In addition, experiences from previous years serve as a feedback source. The Department receives feedback concerning the supply of courses from Faculty. The Department makes a proposal of supply of courses but it is the Faculty that approves the proposal. In addition, experiences from previous years serve as a feedback source as well as representatives of the employers.

Arrow 4	As a result of feedback, a corrective loop may occur. The student returns to adjust the goals according to the provided feedback. For example, reflection of plans may disclose inconsistency of the goals and the plans. Therefore, either the plans, the goals or both need to be adjusted. The corrective loop occurs as the student either decides to spend more time on planning or ~~to~~ on returning to adjust the goals and then to start the planning again.	As a result of feedback, a corrective loop may happen. The teacher returns to adjust the goals according to the feedback. For example, after the plans are made the teacher notices that not all goals can be achieved within the given timeframe. The teacher then returns to the setting goals phase to readjust the priority or emphasis of the goals.	As a result of feedback, a corrective loop may happen. For example, the Council of the Helsinki University of Technology has performance negotiates with the Ministry of the Education concerning the number of the degrees and the size of the appropriation. The goals and plans are readjusted during the performance negotiations. Faculty negotiates with the Council of the Helsinki University of Technology about the number of the degrees and the size of the appropriation. In following years the Department may use the feedback it has got from previous year's planning phase to readjust its goals. For example, the representative of the employer may inform that they would need employees that know how to program using C rather than some other language. This type of feedback concerns the whole instructional process and thus concerns the planning phase, too. As a result of the feedback the Department may specify its goals during the next instructional process.
Arrow 5	The student transition from the planning phase to the studying phase.	The teacher transition from the planning to the teaching/studying phase where the plans become concrete.	The different levels of organisation begin actual realization of the plans.
Teaching/ studying	The student follows the study plans. For example, the student goes to the lectures, does exercises, and studies the books. The studying phase includes all students' activities that aim at attaining the goals.	The teacher acts according to the plans. For example, the teacher gives lectures and discusses with the students. The teaching phase includes concrete teacher's activities that aim to attain the goals that the teacher has set earlier.	The Council of the Helsinki University of Technology sees that there are resources for the teaching, for example, in a form of appropriations, premises, and infrastructure. The university also produces indirect services for the whole university that promote its mission (e.g., library, Computing Center, Student Services, and pedagogical support for teachers). The council also decides on the goals of the degree programmes. The Faculty distributes appropriations further to the Departments. The Faculty makes changes to the content of degree programmes if needed. The Faculty also provides tutor system as well as some orientation type of courses for

			the students as a sort of guidance that promotes high-quality education. The department offers courses for majors and minors.
Arrow 6	The student receives feedback during the studying phase. There are at least three feedback sources: self-reflection, peers and the teacher. For example, as the student is studying he/she has a feeling how he/she is doing. The student asks the following self-efficacy questions: do I understand what the lecturer is saying, am I confident that I have mastered some particular skill, and so forth. Peers are another source for feedback. Seeing how the peers are doing and comparing own achievements to theirs gives feedback how one is doing compared to the others. The teacher and teacher's designed exercises are also a source for feedback. The feedback from the teacher might be direct and explicit. For example, when students are doing exercises the teacher might explicitly say how the student is doing. The grading from exercises, whether automatically created or given by the teacher personally, offers the student feedback on how well he or she is doing in respect to the course official goals.	The teacher receives feedback at the teaching phase. The students and the teacher's observations serve as a feedback source. For example, the teacher may observe students during the lectures and exercise sessions and get some indirect feedback on how well they have understood the lectures and the learning material. On the other hand, the feedback from the students may be direct as well. For instance, the students may ask the teacher to repeat and elaborate certain issues during the lectures.	The different level of organisations obtain feedback during the teaching/studying–phase. The Council of the Helsinki University of Technology receives feedback concerning its actions from the previous years. The Faculty collects feedback from the courses departments offers to the students. Therefore, it receives feedback from students and teachers if there are problems with some courses. The same course feedback can be used both at the Faculty and at the Department level. In addition, comparing the experiences from previous years with present situation serve as a feedback source. As the Department provides the concrete courses there are a lot of feedback available, for example, in a form of course feedback that students give after each course. In addition, the Department follows the number of the students at the courses from the resources point of view.
Arrows 7, 8	As a result of feedback, a corrective loop may happen. The feedback from the teaching/studying phase has many implications. First, it	As a result of feedback, a corrective loop may happen. The feedback that is received can be used to evaluate both the plans and the goals	The Council of the Helsinki University of technology uses the feedback from teaching/learning phase to readjust goals and plans of the next process. Faculty uses the feedback from the previous years to adjust

	may act as an impetus for revising the plans (arrow 7). The feedback may imply, for example, that the student needs to spend more time with the learning material in order to attain the goals set earlier. On the other hand, the feedback and experiences from the teaching/learning phase may imply that the goals need to be readjusted (arrow 8).	of the course. For instance, the feedback from the teaching situation gives some information how well plans are working. According to that feedback, the teacher can go back to the previous phase and the plans can be readjusted if needed (arrow 7). The feedback from the teaching phase also gives some hints whether the goals of the course will be met. As a result, the teacher may return to the goals and readjust the emphasis of the different goals (arrow 8). The essential goals may get even more weight and less focal goals are put into background. However, adjusting the goals may not be possible until the next instructional process starts.	the next instructional process. For example, the Faculty has received feedback that there are many incompatible course timetables. The next year the faculty can improve the timetable accordingly. Department uses the feedback to readjust the plans. For example, the Department may allocate more money to a particular course, for example, if more students decide to enter the course than was anticipated. On the other hand, the Department may use the feedback to readjust the next year's goals and plans.
Arrow 9	The student transition to the phase where the outcome of the studying can be observed.	The teacher transition to the state where the outcomes of the process can be observed.	The different level of organisation transition from teaching/learning phase to observing the outcomes of the process.
Outcome	The outcome of the process becomes visible to the student. There are outcomes that emerge already during the course and others that emerge only after the course or even after years. For example, the student does exercises that are part of the course. The marking of the exercise or the final grade of the course is an outcome that stands for some skill or knowledge that the student has acquired. A single course may have also contributed to the student's attitudes and other less visible characteristics. However, these kinds of outcomes may not unfold until years after.	The outcome of the process becomes visible to the teacher. For example, the teacher sees how the students have acquired themselves from the exercises or the final exam of the course.	The Council of the Helsinki University of Technology observes, for example, how many degrees the Faculties have produced during the year. In addition, the Council may observe the number of the scientific publications that are produced during the year as an indicator of high-quality research. The Faculty observes how many degrees the Departments have produced during the year. The department observes how many study credits and degrees it has produced during the year.

Arrow 10	The student receives feedback at the outcome phase. There are several types of outcomes varying from concrete skills to conceptual knowledge and attitudes. The sources of the feedback are also diverse. For example, the student can compare the outcomes he/she has attained to the goals he/she set for him/herself at the beginning of the course. On the other hand, he/she can compare the outcomes to the goals the teacher had set to the course. The final grade of the course gives the student feedback concerning how he/she did compared with the goals that the teacher had set to the course. The grade enables a comparison with the other students and thus provides the feedback as well.	The teacher receives feedback at the outcome phase. There are several types of outcomes. However, not all of them are visible to the teacher right after the instructional process is over. For example, the teacher may observe the students ability to program or describe some concepts and thereby get feedback how well the goals of the course have been met.	The different levels of organisation get feedback at outcome phase. Alumnae and representatives of industry provide feedback concerning the content of the degree programmes and the knowledge and ability levels of the graduates for the different levels of the organisation. This feedback organisation can use it to advance either certain phase of the instructional process or the whole process as an entity. The feedback is also available in the form of international comparisons between universities. The placement on such ranking lists gives some idea how well the university is succeeded in high international level of research, education, and teaching. Other sources of feedback are the students. The number and the quality of the students that apply for study place reflect partly the respect and the standard of the university. It is likely that if the university has a good reputation of producing qualified professionals that have good employment opportunities, then it will attract applicants. The University also follows the average grades and the number of years it takes from students to graduate. For example, the Council of the Helsinki University of Technology may compare the number of the degrees to the earlier set goals and outcomes of previous years. The Ministry of Education also gives feedback; although, this feedback is not realised before the next year's distribution of appropriations. For instance, the Ministry of Education may decide to cut down the appropriation if the previous year's goals were not met. The Faculty compares the number of the degrees to the earlier set goals. The result of the comparison serves as a feedback that indicates how successfully the faculty attained its quantitative goals. In addition, outcomes and experiences from previous years serve as a feedback source, too. Department compares the number of the degrees and study

			credits with the earlier set goals. The passing percents and average grades of the courses provide the Department the feedback it can use to develop the courses. In addition, after years, the Department receives feedback from the industry concerning the skills of the former students. How well graduates knowledge and skills match the needs of the industry is the type of the feedback that Department needs to develop its courses.
Arrows 11, 12	As a result of the feedback, a corrective loop may happen. As learning is a continuous process, there are outcomes available already during the instructional process. The feedback that concerns those sub outcomes may affect the other phases of the process. For example, the marking of the exercise suggest that more studying is needed. A student returns to readjust his/hers plans (arrow 12). This could mean, for instance, as drastic measures as dropping out some other course in order to make time for the course in question. On the other hand, a student may decide to study the difficult subject more and thus returns to the studying phase (arrow 12).	As a result of the feedback, a corrective loop may happen. As a result of the feedback, the teacher may decide to readjust the plans and teaching. For instance, after seeing how the students acquire themselves from the exercises the teacher perceives that teaching is not promoting the learning he/she was hoping for. Therefore, he/she returns to the planning phase to adjust the plans (arrow 12). He/she might decide, for instance, to devote more time teaching certain concepts. At the teaching phase, he/she might decide to use more simulations to help the students to learn.	The Council of the Helsinki University of Technology, the Faculty, and the Department may use the feedback from the outcome phase to readjust planning and teaching/studying phase during the following instructional processes.
Arrow 13	The arrow 13 has two meanings. Firstly, it stands for corrective loop that may happen as a result of feedback that a student receives at the outcome phase. For example, the outcome (such as exercise points) suggests that the goals need to be adjusted. Secondly, the arrow 13 stands for the transfer from outcome phase to the setting goals for the next course.	The arrow 13 has two meanings. Firstly, it stands for a corrective loop that may happen as a result of feedback that the teacher receives at the outcome phase. The teacher returns to adjust the goals of the instructional process that is happening at the moment. During the instructional process, such as a single course, the teacher may observe the outcomes that are available during the course of the instructional process. The	The arrow 13 has two meanings. Firstly, it stands for corrective loop that may happen as a result of feedback that different levels of organisation receive at the outcome phase. In practice, the possible corrective loops happen during the following instructional processes. Secondly, the arrow 13 stands for the transfer from outcome phase to the setting goals for the following instructional process.

		outcomes from the exercise may, for instance, suggest that the students have not yet understood some central concepts of the course. This may result in readjusting the goals so that the focal goals are emphasised more, at the cost of eliminating the less focal goals. Secondly, the arrow 13 stands for the transfer from the outcome phase to the setting goals for the next course.	
Arrow 14	At the beginning of the new instructional process, a student looks back at the outcome of the previous courses. At the beginning of new course, a student may use the experiences of the previous courses as a base for his/her goal setting. The outcomes of previous courses give a student a feeling for self-efficacy concerning the subject of the course. For example, the student has got good grades from the previous math courses. Therefore he/she is confident about his/hers ability to learn mathematics and sets high goals for the next math course.	At the beginning of the new instructional process, the teacher looks back at the outcome of the previous courses. At the beginning of the new course, the teacher may use the experiences of the previous courses as a base for setting goals for the new course. For example, the outcomes of the previous course suggest that the students had mastered some particular skills well. Therefore, the teacher may conclude that the skill related goals of the previous course were on a suitable level for this student group.	At the beginning of next year's negotiations, the different levels of organisation look back at the outcome of the previous year. The Council of the Helsinki University of Technology, the Faculties and Departments use the number of the degrees relative to appropriations from the previous year to set goals for the next year. For example, Ministry of Education may use the number of degrees the University has produced during the previous year as a motive of the distribution of appropriations for the next year. If the University has not been able to meet the goals of the previous years, the Ministry may cut down the appropriation for the next year. The Council of the Helsinki University of Technology may use the number of degrees the Faculty has produced during the previous year as a motive of the distribution of appropriations for the next year. The outcome functions as a steering mechanism for Departments next year goal setting. For example, the Faculty distributes the appropriation according to the number of the degrees the Department has been able to produce during the previous years.

Appendix 2 Questionnaire for students who had dropped out of the introductory programming course

Background questions

1. Degree program
2. What year are you currently studying in?
3. Student ID
4. Gender
5. I got a grade from the introductory programming course in spring 2005?
    - Yes
    - No
6. Did you get a grade from the summer introductory programming course?
    - Yes
    - No
7. Are you taking some programming course in fall 2005?
    - Yes
    - No
8. Have you planned to include take advanced programming courses?
    - Yes
    - No
9. Is the introductory programming course Y1 compulsory for you?
    - Yes
    - No
10. Did you have friends in the spring introductory programming course?
    - Yes
    - No
11. Which of the following statements describe you the best?
    - I started to do the programming exercises but I did not receive a grade for the exercises
    - I got a grade for the exercises
    - I did not start on the project
    - I started on the project but I did not finish it
    - Other
12. If you chose Other, please specify here
13. Did you have any programming experience before this course?
    - Yes
    - No

If you had no programming experience, you can move to the question 19.

14. Have you ever tried to get a grade from either fall (L1) or spring (Y1) introductory programming courses here at the TKK?
    - Yes
    - No
15. Did you take some programming courses in high-school/applied science university or university?
    - Yes
    - No
16. Which languages have you used for programming?
    - Java
    - C
    - C++
    - Pascal
    - Basic
    - Visual Basic
    - Python
    - Perl
    - Delphi
    - Some other

17. How many lines was your longest program?
18. Was object-oriented programming familiar to you before taking the spring introductory programming course?
    - Object-oriented programming was not familiar to me before taking the course
    - Object-oriented programming was familiar to me because I had tried to get a grade from the course before
    - Object-oriented programming was familiar to me because of some other reason

Lectures

19. The portion of the lectures I attended was approximately
    - 80-100%
    - 60-80%
    - 40-60%
    - 20-40%
    - 0-20%
    - I did not attend lectures

20. If you did not attend lectures or if you attended only few lectures, what was the reason for that?
    - Lectures were held at the same time with other course's lectures
    - I was working at the time when the lectures were held
    - I attended few lectures but I did not learn anything from them

- I did not attend the lectures because I typically do not think that I learn anything at lectures
- I prefer to study the same amount of time independently rather than attend lectures
- I did not bother to go the lectures
- Other reason

21. If you chose ”Other reason”, what was it? (open)
22. If you attended lectures, which of the following statements are correct in your opinion?
    - Lectures discussed topics that helped me to do the programming exercises and to prepare for the exam
    - Lectures did not discuss topics that helped me to do the programming exercises and to prepare for the exam
    - Lectures discussed, generally, essential and interesting topics
    - Lectures generally discussed-unessential and boring topics
    - The lecturer presented the topics in the way that I was able to understand them
    - The lecturer was not able to present the topics in the way that I would have understood them
    - Lecturer reified difficult topics
    - Lectures were too theoretical
23. Slides
    - Slides were clear and informative
    - Slides were difficult to understand
    - I find it important that I can print the slides before the lecture
    - I usually don’t print the slides before the lecture
24. Other comments concerning the slides. (open)

About programming exercise groups and programming exercises

25. Did you attend the programming exercise group?
    - No
    - I attended a few times
    - I attended the group weekly
    - I attended the group several times per week
26. If you did not attend the exercise group or you attended only few times, please pick the suitable option(s)
    - The exercise groups’ schedule was not good for me
    - I did not attend the groups because I wanted to do the exercises alone
    - I did not attend the groups because I did not need any help doing them
    - I did not attend the groups because I had friends that were able to help me if needed
    - It took too much time to get help in the exercise groups
    - Teaching assistants were not able to help me

27. Your thoughts about programming exercises
    - Programming exercises concerned central aspects of the course
    - Programming exercises did not concerned central aspects of the course
    - Programming assignments were clear
    - Programming assignments were unclear

28. If you did not finish programming exercises what was the reason? You may choose more than one option.
    - I did not understand the topics that were covered in the course
    - I did not know how to do the programming exercises
    - I started on the programming exercises too late and I did not have time to finish them by the deadline
    - I did not do them because I wanted to concentrate on other courses
    - I was not able to use as much time for the course as I had planned before.
    - Personal/family reasons
    - Other
29. If you chose ”Other reason”, what was it? (open)

About the project work

30. Which of the following statements describes your opinion?
    - The course (lectures, programming exercises, the text book) helped me to acquire sufficient skills to do the project work
    - The course did not helped me to acquire sufficient skills to do the project work
    - I received enough help to do my project work
    - I would have needed more help
    - The project assignment was well-defined
    - The project assignment was not well-defined

31. About the project work
    - I did not start doing the project work because I did not receive a grade from previous programming exercises
    - I did not start doing the project work because I did not understand the topics that were covered in the course
    - I did not start doing the project work because it felt like it was too large an entity
    - I started doing the project work but I did not know how to finish it
    - I started to do the project work too late and did not have enough time to finish it
    - I estimated that it would take less time to do the project work than it actually did and that is why I did not have time to finish it
    - I did not finish the project because I did not receive enough help and support to do it (e.g., help from exercise groups)
32. What was the hardest about doing the project work? (open)
33. Other comments about the project work. (open)

About the course material and content

34. Which of the following materials you used to study?
    - Slides
    - Some text book
    - Some other material (e.g., web-pages)
    - Educational material provided by the teacher

35. About the content of the educational material that was provided by the teacher
    - The content concerned essential topics of the course
    - The content did not concern essential topics of the course
    - The material was easy-to-understand
    - The material was difficult to understand

36. Other comments concerning the educational material (open question)
37. Which topics that were covered in the course did you find interesting?
38. Which topics that were covered in the course did you find uninteresting?
39. Which topics did you find difficult to understand or learn?
40. News group: Did you read the comments in the course news group? (you may choose more than one option)
    - What news group? I did not know that there was a news group
    - I knew there was a news group but I did not read any messages on it
    - I sought help for my programming assignments from the news group
    - I sought help for my project work from the news group
    - I sought help for some other course related topic from the news group

Study skills and time management

41. Select the one that describes you the best
    - I am usually good at estimating how much time courses take
    - I often underestimate how much time courses take
    - I usually start doing exercises/preparing for exams well in time
    - I usually start doing exercises/preparing for exams at the last moment
    - I know how to adapt different study strategies according to different courses
    - I would like to get more information and guidance about study strategies
42. In January, how many study weeks did you plan to study this spring?
43. How many study weeks did you receive this spring?
44. How many hours per week did you allocate for this course?
45. How many hours did you use before dropping the course?

Reasons for dropping out of the course

46. What was the reason for dropping out of the course? (you may choose more than one option)
    - I did not understand the topics that were covered in the course
    - I did not know how to make the programming exercises

- It took a long time to do the programming exercises and I had to drop out in order to able to concentrate on other courses
- It took a long time to do the project work and I had to drop out in order to able to concentrate on other courses
- I was not able to use as much time for the course as I had planned
- I did not have time since I was preparing for entrance examination
- Personal/family reasons
- Other reasons, please elaborate

47. What would have prevented you from dropping out of the course? (open)
48. Here you may elaborate on your decision for dropping the course (open)
49. Would you like to come for an interview to talk about your experiences in the course?

    If yes, please give your email address.

Appendix 3 An interview plan for students who had dropped out of the introductory programming course

1. Warm up questions that concern issues, such as study year and department
2. At which point of the course you dropped out?
    (a) How many exercise round you did?
    (b) Did you do the project plan?
    (c) If you started to do the programming project, how far you got?
3. Feedback concerning the course content, lectures, study material, programming exercises and project, teaching, and exercise groups.
    (a) What could be done better?
    (b) Did course arrangements affect your decision to drop out?
4. Course content
    (a) What was hardest part to understand?
    (b) What part of the course was most time-consuming?
5. Why did you drop out of the CS1 course?
    (a) If the reason is no time
        i. Why you did not have time?
        ii. Were you able to use the time you had efficiently?
        iii. Was there something that disturbed your studying?
        iv. Do you use some time-managing system?
6. Did you have any friends at the course?
    (a) Do you see the fact that you had friend/did not had friend at the course being relevant/irrelevant?
7. How do you feel like studying at Helsinki University of Technology (TKK)?
8. General feedback

Appendix 4 Questionnaire for both dropped out and passed students

Part I

1. Student ID
2. Gender
3. Have you planned to take advanced programming courses?
4. Was object-oriented programming familiar to you before enrolling in the course?
5. How was your general study motivation this spring? (very low/some motivational problems/neutral (not particularly low or high)/very good)
6. How many credits did you plan in January to get this spring?
7. How many credits did you get this spring?
8. Evaluate the following statements:

a. I usually start doing exercises/preparing for an exam well in time. (true/not true)

b. I am good at estimating how much time a course takes. (true/not true)

9. Which of the following describes you the best?

    a. I am studying what I want to.

    b. I am not quite sure is this what I want to study.

    c. I would like to study something else but I do not know what it could be/I have not been accepted as a student yet.

    d. Other

10. Elaborate if you chose other (open)
11. How do you enjoy studying at this university? Elaborate on why you enjoy/do not enjoy it. (open)

Part II

12. Please rate how difficult it was to (it was not at all difficult/it was difficult at first but got easier soon/it was difficult for a long time but become easier eventually/it was difficult till the end of the course/I do not know)

    - Use a text editor
    - Discover a conceptual solution to the problem
    - Discover an algorithm that executes the conceptual solution
    - Transfer your own thinking into programming language
    - Identify structures in a given code
    - Understand how given code is executed
    - Design the parts of your own code
    - Design the functioning of your own code
    - Find compile-time errors
    - Find runtime errors
    - Test your own code
    - Adopt the exactness needed in writing program
    - Adopt the programming style required in the course

• Mathematics that are needed to solve the problems
• Other

13. Please rate how difficult the following course content-related issues were (it was easy to understand right from the beginning/it was difficult to understand at first but I understood it soon/it was difficult to understand for a long time but I understood it eventually/I did not understand it during the course)

• Conditional statements (such as if-else)
• Loops (such as while)
• The idea of OO
• Methods
• Tables
• Inheritance and abstract classes
• Exceptions
• Handling files
• Other

14. Elaborate if you chose 'other'. (open)
15. What got you through the previously listed difficult issues? (open)

Part III

16. Please rate how crucial the following motivation- and studying-related reasons were to your decision to drop out. (no effect at all/ minor effect/ some effect/ critical effect/ I do not know)

• My general study motivation was low
• Programming does not interest me
• The course workload is not in balance with the payback
• I had no intention to pass the course in the first place.
• The course is not mandatory for me.
• Dropping out does not affect other courses.
• I am not going to stay at this university

17. Please rate how crucial the following time management-related reasons were to your decision to drop out. (no affect at all/ minor affect/ some affect/ critical affect/ I do not know)

• It took too much time to do the programming exercises.
• I had reserved too little time for the 5 ECTS credit course.
• The course took more time than anticipated (more time than other 5-credit courses).
• I had a lot of courses at spring.
• I wanted to concentrate on other courses.
• I decided to use my time to read for an entrance exam.
• Work commitments prevented me from using enough time for the course
• Hobby commitments prevented me from using enough time for the course

18. Please rate how crucial the following reasons were to your decision to drop out (no affect at all/ minor affect/ some affect/ critical affect/ I do not know).

- I did not understand the course content.
- I did not know how to do the programming exercises.
- I did not get enough help
- The course personnel’s actions (elaborate later, question 19)
- The course arrangements (elaborate later, question 19)
- I got caught plagiarising
- Personal/family related reasons
- Other (elaborate later, question 19)

19. Please elaborate if course personnel actions/course arrangements played a part in your decision to drop out or if you chose the option Other. (open)

20. If you chose more than one reason that contributed to your dropout decision, please elaborate if those reasons were connected to each other. (open)

21. How was the course workload compared with the credit points?

22. Rate the difficulty of the course.

23. Here you may elaborate if you have something you would like to say about the CS1 course or studying in general. (open)

Appendix 5 Questionnaire for computer science teachers

1. Name
2. The university where you work
3. Educational background:
    a. Master of Science student
    b. Bachelor of Science degree
    c. Masters of Science degree
    d. Licentiate of Science
    e. PhD
4. My background is in
    a. Computer science
    b. In some other science than computer science, please elaborate on which science
5. I have worked as
    a. a teaching assistance
    b. an assistant teacher
    c. a tutor
    d. a lecturer
    e. I do not have teaching experience
6. My job title is
    a. research assistant
    b. researcher
    c. teaching researcher
    d. lecturer
    e. professor
    f. docent
    g. planning officer
    h. student advisor
7. Pedagogical education
    a. I have taken some larger set of pedagogical courses
    b. I have taken single pedagogical courses or I have studied the pedagogical material by myself
    c. I do not have any pedagogical training
8. Using your own term define what studying is. (open)
9. Elaborate on what kind of aspects are involved in studying? (open)
10. Which aspects of studying are essential for successful studying? (open)
11. Which aspects of studying are difficult for students? (open)
12. Can teacher/assistant teacher affect the student's studying process? If yes/no, please give examples. (open)
13. Is there anything else you would like to tell? (open)

Appendix 6 An interview plan for computer science teachers

1. Tell me about your educational background
2. When the new instructional process, such as one of your own courses, starts, what is the first thing you do?
3. What are the goals of the course?
    a. Content, skills
    b. Goals for CS majors and non CS majors
    c. Do you receive feedback concerning the goals?
    d. Does the feedback affect the goals?
4. How do you plan your course?
    a. Do you receive feedback concerning your plans?
    b. Does the feedback affect the plans?
    c. Have you ever adjusted your goals according the feedback?
5. What do you do when the course is running?
    a. Concrete tasks
    b. Collaboration with teaching assistants
    c. What kind of feedback you receive when you are teaching the course? Who gives feedback? What kind of feedback?
    d. Does the feedback affect your teaching, plans, and the goals of the course?
6. Tell me about the learning outcomes of the course
    a. What kind of outcomes there are?
    b. What do you do with the outcomes?
    c. Do you receive feedback concerning the outcomes
    d. Does the feedback affect goals or plans of the course, or your teaching?
7. How do you support the students studying process in your course?

Appendix 7 An interview plan for the representatives of the administration

1. Educational background
2. Job description
    a. How long has been doing this job
3. Tell me about what kind of goals you have in your job.
    a. Who sets the goals
    b. What the goals mean in a concrete level, give examples, please
    c. Do you receive feedback concerning the goals you have set?
    d. How does the feedback affect your work?
4. How you make plans to achieve the goals?
    a. Who makes the plans?
    b. What does planning mean in a concrete level? What do you do when you plan?
    c. Do you receive feedback concerning your plans? Who gives the feedback
    d. How does the feedback affect your work?
5. How do you implement the plans?
    a. Give concrete examples on how you implement the plans.
    b. Do you receive feedback when you implement the plans? Who gives feedback? What kind of feedback do you receive?
    c. How does the feedback affect your work?
6. After you have implemented the plans, what kinds of outcomes are there?
    a. Give examples of outcomes.
    b. Do you receive feedback concerning the outcomes? Who gives the feedback? What kind of feedback?
    c. How does the feedback affect your work?
7. Other comments?

Appendix 8 Summarised interview data

Student 1, year of study IV

	Prior experience	Plans	Motivation	Social aspects	Studying habits	Course	Timetable	Studying
	Has tried to take the course before but dropped out because of low motivation. Has limited experience with several programming language. Object oriented programming is familiar from previous attempt to take the course	This is a compulsory course. Plans to take advanced programming courses. Plans to get 16 study weeks during the spring semester	Excited, motivated	Does not have friends at the course.	Is usually good at estimating how much time courses take. Usually tends to start doing exercises in the nick of time		Has made a timetable when to do exercises. Plans to use 5-10 hours/week for this course.	
Programming exercises					Attends 40-60% of lectures. Studies educational material. Attends the exercise groups few times. Preferred doing exercises alone. Seeks help from news group.	Programming exercises get more complicated	Exercises require more time. Other courses require time.	Gets stuck on a difficult spot and gets nowhere, takes time to figure out how to get further.
Group exercises			Motivation drops	Hard to find the group where to do group work			Timetable gets messed up	Didn't do the group work exercise, which put a pressure to get good grades from other exercises.
		Is not sure anymore about taking the advanced courses					**Runs out of time and drops out.**	

Student 2, year of study IV+

	Prior experience	**Plans**	**Social aspects/climate**	**Studying habits**	**Course**	**Timetable**	**Studying**
	Has tried to take the course before but had to drop out because the project work was not good enough. Has some experience of programming from high school. Is familiar with object oriented programming.	The course was optional. Aims at passing grade. No plans to take advanced programming courses Plans to get 39 study weeks during the spring semester	Had two siblings at the course at the same time.	Is able to adapt different study strategies according different courses.		Plans to use 5 hours/ week for this course	
Programming exercises			Feels that the most of the course staff is set against students.	Didn't attend lectures because had attended them already at the previous time. Studies course slides and uses internet for source of information. Didn't attend exercise groups because the timetables were incompatible and he/she had brothers that were able to help. Seeks help from news group.	There are some exercises where good style is especially emphasised.		Neglects some exercises because of lack of time. Fails the style test. Feels that the consequences of failing the style test are unfair.
Project work				Is used to do projects with his/her friends. At his/her department it is a norm that students co-operate when doing projects even though it is not officially allowed. Is used to that unauthorised group work is frequently overlooked.			Project assignment was unclear. The course didn't give good enough skills and abilities to cope with the project work. Does parts of the project with his/her sibling.
			Feels that the style with which the plagiarism is announced is very hostile. No clarifications or explanations are asked.	Is upset because there are differences how course staff reacts to unauthorised group work.	**Is accused of plagiarism and as a consequence-has to drop out the course.**		

Student 3, year of study IV+

	Prior experience	Plans	Social aspects	Studying habits	Course	Timetable	Studying
	Has tried to take the course before. Has some experience of programming from University of Applied Sciences. Object oriented programming is familiar from previous attempt to take the course.	The course was optional. Plans to take advanced programming courses.	No friends at the same course. In general, has noticed social discrimination	Is able to adapt different study strategies according different courses.			
Programming exercises				Attended few lectures but didn't think it was beneficial for learning. Preferred studying the equivalent time independently. Attended few exercise groups but it took a long time to get help and sometimes assistant teacher was not able to help. Reeds books and other educational material. Material is obscure, which resulted that it took long time to make sense of it. Seeks help from news group.	Programming exercises didn't cover the essential aspects of programming. Assignments were unclear. Assistant teachers may know the subject but are not able to explain and teach.	It took too much time to do exercises. Other courses required time too.	Neglects some exercises because of lack of time.
Project work			Didn't want to bother course staff with questions because it was not absolutely necessary for him/her to pass this course.	I didn't get enough help to be able to do the project.	Project required some knowledge of algorithms The learning material didn't offer enough support.	I was not able to use as much time for the course as I had planned.	Project assignment was unclear. The course didn't give good enough skills and abilities to cope with the project work.
					Drops out because doesn't know how to finish project.		

Student 4, year of study IV

	Prior experience	Plans	Motivation	Social aspects	Studying habits	Course	Timetable	Studying
	No prior experience of programming.	The course was optional. Aims at getting 25.5 study weeks during the spring semester. No plans to take advanced programming course.	Has personal interest. Aims at gaining more general technical knowledge. Has noticed that knowledge about programming is beneficial at work.	No friends at the same course. Had friend who was able to help if needed.	Usually underestimates how much time courses take. Usually tends to start doing exercises in the nick of time. Is able to adapt different study strategies according different courses.		Plans to use 2 hours/week for this course	
Programming exercises					Attended 20-40% of lectures. Lectures were at the same time with other course's lectures. Studied slides, books, and educational material.	Study material covered the basic information but not enough to do exercises.		
Project work			Motivation drops since the aim is achieved.		I didn't get enough help to be able to do the project.		Other exams and projects at the same time. Prefers compulsory courses. Project would have required a lot of time and wouldn't give any added value relative to the personal aims.	Did the plan for project but didn't start doing it. The course didn't give good enough skills and abilities to cope with the project work.
		Receives 21,5 study weeks					**Drops out to be able to concentrate on other courses.**	

Student 5, year of study III

	Prior experience	**Plans**	**Motivation**	**Social aspects**	**Studying habits**
	No prior experience of programming	The course is optional. Aims at getting 20 study weeks during the spring semester. Plans to take advanced programming courses.		No friends at the same course.	Is usually good at estimating how much time courses take. Usually starts doing exercises/preparing for exams well in advance. Is able to adapt different study strategies according different courses.
Programming exercises		Notices that this is not the course he/she is supposed to take but the parallel programming course.	To get a feel for the course. What kind of exercises there are, how difficult the course is. Does not aim at getting passing grade.		Didn't attend exercise groups. Uses internet or other material not provided by the course for source of information. Seeks help from news group
		Drops out			

Student 6, year of study III

	Prior experience	Plans	Motivation	Social aspects	Studying habits	Course	Timetable	Studying
	Has experience of programming from University of Applied Sciences. Object oriented programming is familiar from previous courses.	The course was optional. Aims at getting 20 study weeks during the spring semester. No plans to take advanced programming course.	Beliefs that this course could be beneficial at work. The course might help to advance logical thinking skills.	No friends at the same course. Prefers going to the course alone without friends. Thinks that can concentrate better that way.	Is usually good at estimating how much time courses take. Usually tends to start doing exercises in the nick of time.		Work commitments.	
Programming exercises					Attended 20-40% of lectures. Lectures were at the same time with other course. Works at the same time. Studies course slides. Attended few times exercise groups.	Programming exercises. It took a long time to get help at the exercise group and sometimes assistant teacher was not able to help.	Other courses require time. Works at the same time. The course has strict deadlines for exercises	Neglects exercises because of lack of time. Didn't get the idea of object –oriented programming. Wasn't able to find the bug in his/her code. Didn't have time to ask anybody.
							Drops out in order to be able to concentrate on other courses.	

Student 7, year of study II

	Prior experience	**Plans**	**Social aspects**	**Studying habits**	**Course**	**Timetable**	**Studying**
	Has limited experience of programming from University of Applied Sciences. Is not familiar with Object-oriented programming.	This is a compulsory course. Plans to take advanced programming courses. Plans to get 15 study weeks during the spring semester	No friends at the same course.	Usually underestimates how much time courses take.			
Programming exercises			The course rule concerning unauthorised group work creates the image that you need to do everything totally alone. Self sees the social aspect very important. Discussions with others help to broaden own mind and promotes the feeling of fellowship and thus prevents withdrawal.	Attended 20-40% of lectures but didn't think it was beneficial for learning. Attended exercise groups several times/week. It took a long time to get help at the group. Needed help but didn't get it. Studies slides, book, and educational material.	Lectures didn't cover the essential aspects concerning the requirements for passing. Programming assignments were unclear. Exercise groups were restless. Assistant teachers were in a hurry. Study material had the basic information but it didn't help with more complicated issues.	It took a long time to do programming exercises. Other courses required time.	I didn't know how to make programming exercises. Neglected the exercises to able to concentrate on other courses.
Group exercises			Didn't work out due to the short time. Forming took time and the group didn't really get to work efficiently.		Group work assignment		
		Gets 5 study weeks				**Drops out to be able to concentrate on other courses.**	

Student 8, year of study II

	Prior experience	Plans	Motivation	Social aspects	Studying habits	Course	Timetable	Studying
	Some prior experience of programming.	This is a compulsory course. Plans to take advanced courses. Plans to get 15 study weeks during the spring semester	Aims at good grade, not just passing the course.	No friends at the course. Is more mature than other students in average. In general, does not fell like fitting in with 19-25 year olds male students. In general, feels "stupid" at the university even though his/her high school grades were the best possible.	Usually tends to start doing exercises in the nick of time.		Has family obligations and several other courses but is good at arranging time for courses.	
Programming exercises				Teaching assistants are immature and don't have social skills.	Attended 60-80% of lectures. Studied slides and seek information from internet. Sought help from news groups. Attended exercise groups several times/week. It took a long time to get help, which was frustrating.	Programming exercises were sometimes obscure. Assistant teachers may know the subject but are not able to explain and teach. Exercise groups were restless.	Wasn't able to use as much time for the course as had planned at the beginning.	Started doing programming exercises too late and wasn't able to finish some of them by deadline.
Project work			It stared to look like a good grade was not possible to get.				Project work took too much time.	Started doing project work too late and wasn't able to finish it by the deadline.
							Drops out to be able to concentrate on other courses.	

Student 9, year of study IV+

	Prior experience	**Plans**	**Motivation**	**Social aspects**	**Studying habits**	**Course**	**Timetable**	**Studying**
	Had limited experience in programming before the course. Is not familiar with Object-oriented programming.	The course was optional. Aims at getting 19 study weeks during the spring semester. No plans to take advanced programming course.	Wanted to learn basics from programming on a level that one knows what people are talking about.	Had friends at the same course. In general likes if there are friends at the same course. Friends push each other and may have a friendly competition among each other.	Usually underestimates how much time courses take.		Plans to use 8 hours/week for the course	
Programming exercises				Friends were able to out with small difficult spots that otherwise would have taken a long time to overcome. Discussions help to understand new information. It is comforting feeling that you are not alone but others struggle with same problems.	Attends 80-100% of lectures. Studies slides, book, educational material, seeks help from other material like internet. Attended exercise groups several times/week Seeks help from news group.	Slides were obscure.	Uses 12 hours/week for the course. Had other compulsory courses at the same time.	I didn't know how to make programming exercises. Neglected the exercises to able to concentrate on other courses.
Project work							Other exams and projects at the same time. Prefers compulsory courses.	Didn't start project because it felt like too large entity.
							Drops out to be able to concentrate on other courses	

Student 10, year of study III

	Prior experience	**Plans**	**Motivation**	**Social aspects**	**Studying habits**	**Course**	**Timetable**	**Studying**
	Has tried to take the parallel course before. Has limited experience of programming. Object oriented programming is familiar from previous attempt to take programming course.	This is a compulsory course. No plans to take advanced programming course. Plans to get 27 study weeks during the spring semester.	Beliefs that can get the grade if works hard.	Had friends at the same course.	Often plans to take more courses than is able to handle. Usually underestimates how much time courses take. The general planning of studies has not been goal oriented until now. Has had to take many basic courses several times to get a passing grade. Tends to study periodically.		Plans to use 10 hours/week for the course	
Programming exercises			Motivation drops as usually after couple of weeks.	Asked friends if got stuck on a difficult spot.	Attended 0-20% of lectures. Lectures were at the same time with other courses. Studies slides and educational material. Didn't attend exercise groups because had friends who were able to help if needed.	Didn't understand the idea of object-oriented programming	Uses 20 hours/week for the course. Programming exercises take too much time to complete.	Neglected the exercises to able to concentrate on other courses (basic courses).
		Gets 12 study weeks during the spring semester					**Drops out to be able to concentrate on other courses.**	

Student 11, year of study II

	Prior experience	**Plans**	**Social aspects**	**Studying habits**	**Timetable**	**Studying**
	No prior experience in programming.	This is a compulsory course. Plans to take advanced programming course. Plans to get 19 study weeks during the spring semester.	Had friends at the same course.	Usually tends to start doing exercises in the nick of time.	Has a lot of other activities. Fraternity and scouts related activities take time.	
Programming exercises			It was nicer to go to the lectures with the friend.	Attends 60-80% of lectures. Studies course slides and uses other material like internet for studying. Attended exercise groups several times/week. Seeks help from news group.	Uses 10 hours plus time spend on lectures/week for the course. Wasn't able to use as much time for the course as had planned at the beginning.	Started doing programming exercises too late/at the last possible day and wasn't able to finish some of them by deadline.
Project work		Had some free time in a following summer since didn't have summer job for the whole summer brake. Plans to take the programming course in the summer.		Would have liked to attend the exercise groups when doing the project. Groups would have offered a source for help.	There were a lot of activities and other courses that took time from the course.	Starting the project was the most difficult. Especially to figuring out how it is best to start to do it. Started doing project work too late and wasn't able to finish it by the deadline.
		Gets 14 study weeks during the spring semester			**Drops out to be able to concentrate on other course and activities.**	

Student 12, year of study IV+

	Prior experience	**Plans**	**Motivation**	**Social aspects**	**Studying habits**	**Course**	**Timetable**	**Studying**
	No prior experience in programming Has tried to get passing grade from this course five times. Had always dropped at the first part of the course. Had noticed that this course requires more time than other courses with the same extent (same amount of study weeks).	This is a compulsory course. No plans to take advanced programming course. Plans to get study 28 weeks during the spring semester. Is about to finish f his/her studies. Starts to do his/her masters thesis next fall semester.	In general, is not interested in programming.	No friends at the same course.	Took the first three years of studying easy and started to put effort on studying after that. Is usually good at estimating how much time courses take. Usually tends to start doing exercises in the nick of time.		Plans to use 6-8 hours/week for the course. Has other courses at the same time.	
Programming exercises			Feels like programming is dull. It takes too much effort to get simply program to work. For example, it took 10 hours to get a program that calculates the average of three numbers to work. Gets frustrated.		Attended 40-60% of lectures. Studies course slides and reads book. Lectures covered aspects of programming that were irrelevant. Attended lectures but didn't feel like that was learning anything. Attended exercise groups few times. It took a long time (30 minutes) to get help at the group. Teaching assistants were not able to help.	Some of the programming assignments were obscure.	Wasn't able to use as much time for the course as had planned at the beginning. Uses 6 hours/week for the course.	Didn't understand the idea of object-oriented programming. Didn't know how to do programming exercises. Started doing programming exercises too late and wasn't able to finish some of them by deadline. Neglected the exercises to able to concentrate on other courses. Does not get a grade from programming exercises

Project work					Didn't start the project because it felt too extensive entity.			The course didn't give good enough skills and abilities to cope with the project work. Would have needed more help with the project. Didn't start doing the project because didn't understand issues that were covered during the course. It felt difficult to put the skills into practice.
		Gets 18 study weeks during the spring semester					**Drops out to be able to concentrate on other courses.**	

Student 13, year of study II

	Prior experience	Plans	Social aspects	Studying habits	Course	Timetable	Studying
	No prior experience in programming.	This is a compulsory course. Plans to take advanced programming course. Plans to get 8 study weeks during the spring semester.	Had friends at the same course.	Is usually good at estimating how much time courses take. Usually starts doing exercises/preparing for exams well in advance.		Hadn't set timetable for this course.	
Programming exercises				Attends 0-20% of lectures. Attended some lectures but didn't feel like that was learning anything. Studies slides, reads book and educational material. Didn't attend exercise groups because didn't live at the campus and had friends that were able to help. Seeks help from news group.	Study material had the basic information but it didn't help with more complicated issues.	Uses 5 hours/week for the course.	Gets grade from the exercises. Gets grade from the exam.
Project work		Is planning on applying on other University. Does not need a grade from the course if is accepted to another university.				Reading for entrance exams of another university takes time.	Would have needed more help with the project. Starts doing the project but it took too much time.
		Gets 3 study weeks during the spring semester				**Drops out to be able to concentrate on other courses and preparing on entrance exams.**	

Student 14, year of study III

	Prior experience	Plans	Motivation	Social aspects	Studying habits	Course	Timetable	Studying
	No prior experience in programming.	This is a compulsory course. Plans to take advanced programming course. Plans to get 15 study weeks during the spring semester.	In general has a low study motivation because feels like that is studying at the wrong department.	Had friends at the same course.	Is usually good at estimating how much time courses require. Usually starts doing exercises/preparing for exams well in advance. Is able to adapt different study strategies according different courses.		Plans to use 8 hours/week for the course.	
Programming exercises			No motivation. The course started to irritate.		Attends 0-20% of lectures. Preferred studying the equivalent time independently. Reads a book. Attends exercise groups few times but prefers doing exercises alone. Seeks help from news groups.	Some of the assignments were obscure. Objects were difficult to understand.	Used about 8 hours/week for the course. Other courses Feels that from the beginner's point of view the time needed for the course equals closer to the 10 than 5 study weeks. Course requires more time than other courses with the same amount of study weeks.	The firs couple of programming exercise rounds were easy but then they got more difficult. Programming exercises took a lot of time to complete.
		Gets 8 study weeks during the spring semester.					**Drops out to be able to concentrate on other courses.**	Decides to study some programming by him/herself before enrolling the course again.

Student 15, year of study IV+

	Prior experience	Plans	Motivation	Social aspects	Studying habits	Course	Timetable	Studying
	Has tried to take course before. Had limited experience of programming.	This is a compulsory course. No plans to take advanced programming course. Plans to get 28 study weeks during the spring semester.		No friends at the course. Does not live at the campus.	Is usually good at estimating how much time courses require. Usually starts doing exercises/preparing for exams well in advance	The course had changed a little bit from the previous version the student had taken four years ago.	Thought that the course wouldn’t take so much time since he/she had some experience on programming due to the previous attempt. Has several other courses and hobbies.	
Programming exercises				.	Attends 40-60% of lectures. Attended some lectures but didn’t feel like that was learning anything. Studies slides, reads book, educational material and other material like internet pages. Didn’t attend the exercise groups. Preferred doing exercises alone. Seeks help from news group.	Felt like the course advanced fast. Assignments were sometimes obscure. Gets irritated that educational material doesn’t give straight answers to the problems that occur when doing programming exercises.	Uses 4-5 hour/week for the course Wasn’t able to use as much time for the course as had planned at the beginning.	Doing programming exercises took too much time. Neglected the exercises to able to concentrate on other courses

Group assignment			The group work assignment results the lost of motivation. In general, does not see the point of doing group works. Gets irritated about the group work since it restricts the time and place where studying takes place. Does not see the benefits of group work.	Feels like the group work assignment restricts his/hers independency to choose where and when to study			Feels that it is burdensome to find compatible timetable with other students to do group assignment.	Didn't do the group work exercise, which put a pressure to get good grades from other exercises.
		Dropping out from the course doesn't have consequences.				Notices that there are still a lot to do on a course that requires time.	Other exams and projects at the same time.	Didn't get a grade from programming exercises.
		Gets 23 study weeks during the spring semester.					**Drops out to be able to concentrate on other courses.**	

Student 16, year of study IV+

	Prior experience	Plans	Social aspects	Studying habits	Timetable	Studying
	Has extensive experience on programming.	This is a compulsory course. Plans to take advanced programming course. Plans to get 12 study weeks during the spring semester.	No friends at the same course. Is older than the most of the students. In general, feels like accepted by fellow students.	Is usually good at estimating how much time courses take.	Plans to use 6 hours/week for the course. Has a job and family commitments.	
Programming exercises				Does not attend lectures because works at the same time. Studies slides, reads book, and used other material like internet pages. Seeks help from the news group. Didn't attend exercise groups because didn't need help with the exercises.	Uses 4 hours/week for the course. Working makes it difficult to concentrate on large course in general. Work and family commitments makes it difficult to find compatible timetable for group work. Had to leave for a long work trip in a middle of the course.	Didn't get a grade from the exercises.
		Gets 0 study weeks during the spring semester.			**Drops out because of work trip.**	

Student 17, year of study III

	Prior experience	**Plans**	**Motivation**	**Social aspects**	**Studying habits**	**Course**	**Timetable**	**Studying**
	No prior experience in programming.	The course is optional. Plans to take advanced programming courses. Aims at getting 25 study weeks during the spring semester.	Chooses programming course over basic computing courses because wants to learn something new. Beliefs that programming skill will be beneficial when applying for a job.	Had friends at the same course.	Usually tends to start doing exercises in the nick of time. Usually underestimates how much time courses take. Is able to adapt different study strategies according different courses.			
Programming exercises					Does not attend any lectures because beliefs that learns programming only by programming. Studies slides, reads book and educational material Attended exercise groups few times. Seeks help from news groups.	Programming exercises. Gets a grade from programming exercises.		Was surprised how much time programming exercises required.
Project work				Since didn't attend lectures it was unclear for him/her how much it was allowed to co-operate with other students.		Project work	Other exams and projects at the same time.	Was surprised how much time project work required. It was difficult to figure out the algorithm that was needed for the project work. Does project work together with a friend.
		Gets 20 study weeks during the spring semester.				**Gets caught of plagiarism and has to drop out the course.**		