

HOW TO INCREASE THE STUDENTS' DEGREE OF INVOLVEMENT AND PARTICIPATION IN PROJECT BASED COURSES?

Diana Saplacan, Kristianstad University

e-post | diana.saplacan@hkr.se

Marijana Teljega, Kristianstad University

e-post | marijana.teljega@hkr.se

How to increase the students' degree of involvement and participation in project based courses?

Abstract: The degree of students' involvement and participation in non-examinatory parts of courses at higher level, such as lectures, is often significantly lower than in the examinatory parts. This can especially be seen in project based courses, where the main examination form is done through the application of theory into a practical project, rather than through a written examination. More specific, this paper focuses on two such courses, from the area of software engineering. The courses are given on the second, respectively third year, of the Bachelor Programme in Software Development, at Kristianstad University. These are also part of the academic loop¹ of the programmes. Here, the lectures were replaced by seminars based teaching, with the purpose of raising the students' degree of active- involvement and participation into the courses, ensuring a better learning quality, enabling more interaction between the groups of students, and finally facilitating a higher application of the theory into the practical project. Through concrete examples, the paper will showcase the changes that were made in these two courses, with regard to the learning outcomes, the academic loop, and formative examination forms, such as seminars, with the purpose of increasing the quality of teaching and learning. Finally, the immediate positive effects of these changes show we need to look further on how we can find more innovative ways of teaching and learning, especially in the project based courses, in order to increase student's degree of involvement and participation.

Keywords: project based courses, seminars, software engineering, academic loop, and pedagogical approaches

¹ Academic loop – “aims to integrate academic knowledge and competences in the main subject fields, and this way to support students' development through the whole education”, Swedish Higher Education Authority 2011–2014, Kristianstad University, Main Field: Computer Science, Evaluation Case: 411-52-14, 411-312-13, Degree: First Cycle, p.8



1. Introduction

1.1 Background Information

The digital era provides us with presentation tools and interactive virtual rooms, but at the same time a lot of focus goes on the information communication technology (ICT) used. This may also often imply that less focus goes to the actual way of presenting the course material and on the teaching methods. A UNESCO report, since 2009, on trends in global higher education, shows that “*lecturing about declarative knowledge can no longer be the default method*” of teaching, and that the teaching paradigm has shifted from a student-oriented approach to a more teacher-oriented approach, where a good teaching should also include the active participation of students in the learning process (Altbach, Reisberg, & Rumbley, 2009).

1.2 Problem Statement

Problem based learning (PBL) in project courses at higher level often covers a wide area of complexity, competences and skills, by combining theory with practice. It also encourages *lifelong learning* and *double-loop learning*. However, among the challenges encountered in such courses are:

- The students do not always apply the knowledge taught in lectures on their project work, i.e., remaining often on a *single-loop* and therefore the assessment might become more difficult.
- Further, the degree of students’ involvement and participation in non-examination based parts of courses at higher level, such as lectures, is often significantly lower than in the examinatory parts.
- The projects shall also be anchored in the research, where the student shall take a critical approach to his or her solution. This becomes problematic especially in software engineering projects, where the student deals with complex software systems which requires advanced technical skills, team work, highly developed communication skills, but also a deep understanding of the working process, and finally of the end user.



How to increase the students' degree of involvement and participation in project based courses?

In this way, a gap between the learning outcomes of the course syllabus and the knowledge taught in lectures was observed, and its practical appliance. This, in its turn, affected also the students' performance in the subsequent courses.

2. Aim and Purpose

This paper showcases the challenges in project based courses, and the solutions to those through two case-studies: two such courses, from the area of software engineering, given on the second, respectively third year, of the Bachelor Programme in Software Development, at Kristianstad University are presented here. The courses are also part of the *academic loop* (Swedish Higher Education Authority 2011-2014, Kristianstad University, 2015) of the programmes.

Through concrete examples, the paper presents the changes that were made in these two courses, with regard to the learning outcomes, **research**, the academic loop, educational frameworks, and formative examination forms, such as seminars, in order to improve the quality of teaching and learning, but also to increase the students' degree of involvement and participation into the course.

Finally, an evaluation of the results has been made, partially through the use of ZEFsurvey² evaluation tool, through itslearning³ learning platform, and finally through Evasys⁴ automated course evaluations feedback system. However, we will limit ourselves to exemplify and discuss the evaluation made through ZEFsurvey, as it illustrates very clear where exactly the lacks are and what we shall take action onto. The immediate observations that could be made, based on the changes made, show that there is a need of looking further on how we can find more innovative and strategical ways of teaching and learning, especially in the project based courses. It also shows that we need to use the information and observations that we have in order to develop further our way of teaching, and the students life-long learning.

²ZEFsurvey is an evaluation tool for data driven decision based making. Link: <http://www.zef.fi/en/>

³Itslearning is a global learning platform used in development, distribution and support of e-learning, also used at Kristianstad University. Link: <https://hkr.itslearning.com/index.aspx>

⁴Evasys is an automated course evaluation feedback system, also used at Kristianstad University. Link: <http://evasys.hkr.se/evasys/indexeva.php>



2.1 Themes

The themes covered in this paper are:

- The relation between theory and practice in project based courses
- Challenges in project based courses when projects shall be based on research
- Students' degree of involvement and participation in project based courses

2.2 Research Questions

RQ1: What are the exact problems we encounter and how can we turn those into wins?

RQ2: How can we improve our teaching approaches with other pedagogical forms in order to stimulate the students' involvement and participation in project based courses?

2.3 Limitations

This paper focuses only on two project courses in the field of software engineering, from the Software Development and Engineering Programme at Kristianstad University and does not discuss other project courses within- or external to the programme. Further, this paper will not present in depth the CDIO⁵ standards, or Demola⁶ project. The reasons these two educational frameworks are shortly presented are: to make an overview on how they are related to the course, and to illustrate that both have positive impact on the course. To find more about CDIO and Demola, see www.cdio.org and www.demola.net. Further, only the evaluation results using ZEFsurvey are presented here, as these will be further used for developing the quality of these courses.

3. Methods

The research methods employed were: literature review, empirical method, quantitative methods through surveys and, evaluation of those. All of these are discussed below.

⁵ (Worldwide CDIO Initiative, 2015) – www.cdio.org

⁶ (Demola, 2015) – www.demola.se



3.1 Summary of Literature Review

On variation theory. In a student-oriented approach, the focus shall be on student's learning, and on ways of developing their skills and abilities through *variation*. According to M. Elmgren and A-S. Henriksson, "*interaction with others is an important component in the active learning*" (sw.: "*interaktion med andra är en väsentlig component av det aktiva lärandet.*") (Elmgren & Henriksson, Vad gynnar lärande?, 2010). The two courses approached in this paper take into consideration the interaction between students, such that the structure of the course is focusing on facilitating student's active learning through *seminars* and *project meetings*, and consequently *formative examination* forms.

On different learning styles. The literature shows that there are different learning styles. For instance, Kolb's theory brings into discussion four learning styles that follow a circular process, as it follows: *concrete experience*, *reflective observation*, *abstract conceptualization* and *active experimentation*. According to Kolb, all these styles should be included in the corkscrew learning process, for a lifelong learning (Kolb & Kolb, 2005) (Ljungblom & Norberg, 2012) where the information intake shall be done through **visual**, **auditory**, **read and write**, and **kines-thesis** senses (VARK) (Hedin, 2006). This learning process is common in problem based learning (PBL) (Gjerde, 2013), and therefore also in project based courses. Further, Kurt Lewin's theory approaches the relation between learning, behavior and environment, and states that learning is "*a result of changed cognitive structures*" (Granberg & Ohlsson, 2009). Thus, in the context of a project based course, the learning is challenged through two main aspects: the project to be done and the group dynamics that may enhance the learning experience, or it may diminish it.

On deutro- and proto-learning. Besides the previous mentioned theories, there are also other theories, such as Ashby's and Bateson's, Argys' and Schön's. R. Ashby discusses in his research the learning through feedback and adaptive behavior, where he refers to learning as a system or organism that can be either stable or unstable, depending on one's *trial* and *error*. This is also referred as *feedback loops* by (Granberg & Ohlsson, 2009). Bateson's theory resonates on *how the learning occurs*, e.g. *proto-learning* and, the process of learning something fast,



e.g. *deutro-learning*. Software engineering courses challenge often both *proto-learning* and *deutro-learning* through the technologies employed and the *PBL*.

On theories of action and theories-in-use. Further, C. Argyris' and D. Schöns' theories relate to *theories of action* and *theories-in-use* (Granberg & Ohlsson, 2009). The former theories present how one acts and behaves for reaching his or her goals and it reflects *single-loop learning*. The latter refers to what one actually does for reaching his or her goal, how the individual understands his or her- and others behaviors, as well as the context, in order to improve it. This mirrors a more complex process of *double-loop learning*, as described by Schön: *knowing-in-action* and *reflecting conversation*. (Granberg & Ohlsson, 2009) Project based courses, in this case software engineering courses, come as a symbiosis between *theories of action* and *theories-in-use*, and therefore imply *double-loop learning*, where the theory interferes with practice.

On seminar teaching approach creating a suitable study environment in project based courses.

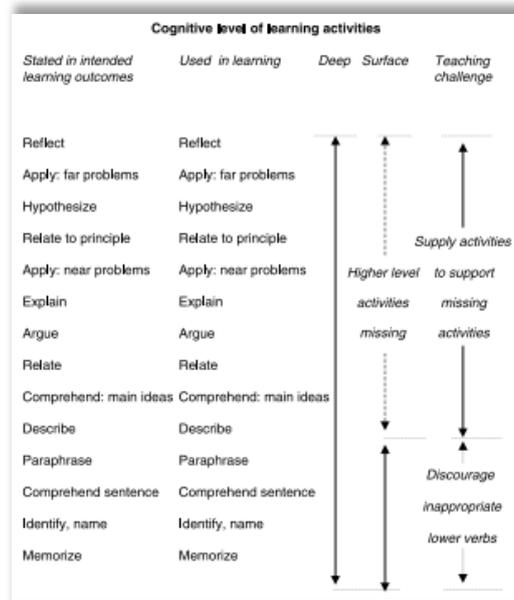


Figure 1. Different level of engagement



How to increase the students' degree of involvement and participation in project based courses?

Different theories have been discussed when planning how to increase students' attendance. Two learning approaches, *surface* and *deep approach*

(Biggs & Tang, 2007), are employed by students. *Surface* approach is connected to “*cutting corners*”, and “*sweeping under the carpet*”, while *deep learning approach* correspond to students' motivation and engagement to learn more deeply. It is also discussed what teachers can do to help students achieve deep learning: in contrast to only presenting theories during lectures, students should be actively engaged, for example through discussions. Moreover, *surface* approach is usually employed when only memorizing theory, whereas *deep approach* when trying to get deeper understanding and meaning (Elmgren & Henriksson, Examination, 2010). Both teacher and students should work towards *deep* approach. Figure 1 shows different levels of engagement for both teachers and students.

Further, SFS's survey results and viewpoints were taken into account when discussed what teaching approach to adopt, which will increase students attendance (SFS - Sveriges förenade studentkårer, 2013): “*We need to be more active in our learning instead of waiting for teachers serving out knowledge*” (*sw.:* “*Vi behöver bli mer aktiva i utbildningen istället för att få kunskap serverad*”).

Another study made at Uppsala University shows that 80% of the asked students think that it is important or highly important that teachers plan and create student-active-learning- environment, for example giving student possibility to ask questions during the lectures (Hedin, 2006). Moreover, it is important for students learning process, to have learning outcomes clearly stated (Elmgren & Henriksson, Examination, 2010) and to give students feedback on the performed work (SFS - Sveriges förenade studentkårer, 2013). Through writing specific and concrete learning outcomes, students are able to understand what goals need to be achieved, which leads to taking responsibility for studies (Elmgren & Henriksson, 2013). Students from both software engineering courses have different backgrounds, and different learning styles, therefore it is important to have that in mind when planning the courses. Different learning styles are neither good nor bad (Elmgren & Henriksson, 2013): a student's learning style depends on inheritance, background and requirements from the surrounding environment. In order to focus to create a varied teaching environment, and consequently to facilitate learning, it is indicated to create groups with different learning styles.



3.2 Educational Frameworks

On one hand, reports and studies show that the quality of *Higher Education* shall be improved with regard to research anchoring (Adamson, 2013) (Säljö & Södling, 2006), and focus on the student's learning (SFS - Sveriges förenade studentkårer, 2013). This is applicable even in the case of the two software engineering courses. Challenges imply not only technical aspects, but also ethical and social aspects that shall be taken into consideration when integrating research into project courses.

On the other hand, the structure of the observed learning outcome (SOLO-taxonomy) shows that the performance and competence of students shall be viewed in regard with the intended learning outcomes of the course, programme, and institution (Biggs & Tang, 2007) (Elmgren & Henriksson, Examination, 2010). In the case of software engineering courses, the SOLO-taxonomy is grounded in the *academic loop* of the programme which “*aims to integrate academic knowledge and competences in the main subject fields, and this way to support students' development through the whole education*” (Swedish Higher Education Authority 2011-2014, Kristianstad University, 2015).

Constructive alignment approach (Biggs & Tang, 2007), regarding learning outcomes, teaching/learning activities and examination, had its focus on achieving flexibility when changes might arise through the course (Elmgren & Henriksson, 2013). Further, The Swedish Higher Education Law (lagen.nu, 1992) states that both students and teachers shall try to obtain quality with focus in science, in both learning-, and teaching activities, as well as planning for future work (lagen.nu, 1992). Swedish Higher Education Ordinance (Högskoleförordningen 1993, 1993) states that university shall give students the opportunity to give feedback during or/and after the course ends. This was adopted in both courses, but in this paper, we will especially focus on the feedback received during Software Engineering DA216A, with the aim of developing further even Software Engineering 2 DA540B during fall-term 2015.

Bologna reform was introduced in the higher educational systems since 2007. It focuses on the importance of approaches of how to examine learnings outcomes. Apart from the Bologna process from which the Swedish Higher Education is part of, Kristianstad University is a member of CDIO framework. CDIO is an interna-



How to increase the students' degree of involvement and participation in project based courses?

tional educational framework, describing CDIO standards stated by technical industry to meet the needs in knowledge of graduated students. CDIO stands for different phases that exist when working in project: Conceive-Design-Implement-Operate. Since 2013 Software Engineering course learning outcomes are connected to CDIO syllabus, but it was also part of Demola, an innovational platform created for students, with its background and developed concept in Finland (Demola). The course project had a positive result – see (Einarson 2012).

3.3 Empirical study

The current research is based on an ongoing quality process, which is also part of the *software engineering track* of the *academic loop*⁷. Based on an evaluation made in 2013 by Swedish Higher Education Authority, the programme in Software Development and Engineering showed that students were lacking quality on the following main aspects:

- Main area of study, scientific basis, applicable methods in the area of study, deep knowledge in a specific part of area of study, as well as relevant current research questions
- To search, collect, evaluate and critically understand relevant information of a problem, as well as discussing phenomena, issues and situations
- To demonstrate the ability to present and discuss information, problems and solutions in dialogue with different groups, both orally and in writing
- To demonstrate the ability in the main field of study to make judgements with consideration to relevant scientific, social and ethical aspects
- (Swedish Higher Education Authority, 2015)

In order to improve those through the whole programme, the *academic loop* was introduced (Swedish Higher Education Authority 2011-2014, Kristianstad University, 2015). In the next section, section 4. *Presentation of Case Studies*, we will look closer to specifically the courses approached in this paper: Software

⁷ The academic loop of the Software Development and Engineering Programme at Kristianstad University has four main tracks: (1) Programming, (2) Mathematics/Algorithm, (3) Software Engineering, and (4) Applied Computer Science. (Swedish Higher Education Authority 2011-2014, Kristianstad University, 2015)



Engineering DA216A⁸ (before Software Engineering I DA523B⁹) and Software Engineering 2 DA540C¹⁰ (before: Software Engineering II DA540B¹¹).

Based on the feedback received from the Swedish Higher Education Authority, the *academic loop*, course syllabus and its intended learning outcomes, as well as through observation, a self-evaluation process and discussions with colleagues at the Department of Design- and computer science at Kristianstad University, new teaching forms were introduced as the main form of teaching in software engineering courses. These consists of: seminar based lecturing where the lecture sessions are divided into two parts: *lecture and seminar*, *log book* to document the project processes and in order to facilitate the assessment process, as well as combining *opposition* and *presentation techniques*. Further these methods were tested, and finally evaluated. Sections 4. *Presentation of Case Studies*, and 5. *Results* present details on this.

3.4 Mixed Study

In the current evaluation process of the two case studies, we have taken into consideration multiple feedback sources, in the form of surveys: at the beginning, middle and at the end of each course. There are however some slightly small differences between the evaluations on the two courses, as the evaluation process is currently an ongoing process. For simplicity reasons, we chose to show an overview of the evaluation methods in the two courses in the table 1.

⁸ Course syllabus Software Engineering DA216A. Link:
<http://www.hkr.se/sv/utbildningar/kurssida/?cCode=DA216A&view=Plan&version=9090>

⁹ Course syllabus Software Engineering I DA523B. Link:
<http://www.hkr.se/sv/utbildningar/kurssida/?cCode=DA523B&view=Plan&version=8304>

¹⁰ Course syllabus Software Engineering II DA540C. Link:
<http://www.hkr.se/sv/utbildningar/kurssida/?cCode=DA540C&view=Plan&version=9061>

¹¹ Course syllabus Software Engineering 2 DA540B. Link:
<http://www.hkr.se/sv/utbildningar/kurssida/?cCode=DA540B&view=Plan&version=7701>



How to increase the students' degree of involvement and participation in project based courses?

Table 1. Overview of the Evaluation Methods in Software Engineering DA216A and Software Engineering 2 DA540C

	Software Engineering DA216A (on the second academic year, undergraduate programme)	Software Engineering DA540C (on the third academic year, undergraduate programme)
Feedback at the beginning of the course	X	X
Mid-course feedback	X	X
Final course survey using ZEFsurvey evaluation method and system	X	N/A
Final course survey using Evasys	X	X

4. Presentation of Case Studies

4.1 Software Engineering DA216A – 7.5 ECTS

This course is given on the second year, second term of the Bachelor Programme in Software Development and Engineering. The general entry requirements for this course are: Object Oriented Programming 7.5 ECTS and Database Technique or equivalent course. The course has the following examinatory parts:

- **Part 1:** (0,5 hp) *Software Development. Labs.* Expected Learnings Outcomes 1, 2, 5-7 and 10 are examined through individual hearings. Grades Fail and Pass are applied.
- **Part 2:** (4,5 hp) *Software Development. Project.* Expected Learnings Outcomes 1-11 are examined in group through project meetings and mutual presentation of the final project. Grades are given per group; U for approved, or 3, 4, 5.



Part 3: (2,5 hp) *Software Development, Seminars*. Expected Learnings Outcomes 1-3, 6, 8, 10 and 11 are examined through active attendance in seminars. Individual grades: U for fail, G for approved

Further, we consider the following parameters: **1)** the programme syllabus with its intended learning outcomes (Kristianstad University, The Education Board for Health and Society, 2013); **2)** the Software Engineering DA216A course syllabus (Kristianstad University, The Education Board for Health and Society, 2015); **3)** CDIO syllabus (Crawley, Lucas, Malmqvist, & Brodeur, 2011). Considering these, we have matched the parameters against each other, as it follows:

Software Engineering DA216A matched against the *programme syllabus* based on four main areas:

- Scientific basis and software development
- Team work
- Planning and time frames
- Discussion with other groups and ethical aspects

- I. Software Engineering DA216A matched against the *CDIO syllabus*:
- Disciplinary knowledge and reasoning
 - Personal and professional skills and attributes
 - Interpersonal skills: teamwork and communication
 - Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context



How to increase the students' degree of involvement and participation in project based courses?

Table 2 illustrates the three parameters matched against each other.

Table 2. Software Engineering DA216A matched to Degree learning outcomes and CDIO syllabus

Degree learning outcomes ¹²	Course intended learning outcomes ¹³ (*2) matched to Degree learning outcomes and CDIO ¹⁴ (*3) syllabus	
Knowledge and understanding	Knowledge and understanding	
show knowledge of the scientific basis of computer technology	to demonstrate the general knowledge on software development (1)	Disciplinary knowledge and reasoning
and its proven know-how and familiarity with topical research and development work, and	to demonstrate software development knowledge in project, within given timeframes (2)	Disciplinary knowledge and reasoning; Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
show broad proficiency in computer technology and	to explain scientific background for software engineering by using adequate words and concepts (3)	Personal and professional skills and attributes; Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
relevant knowledge of mathematics and natural sciences		
and an ability to communicate this in English		
Skills and abilities	Skills and abilities	
show an ability to independently and creatively identify, formulate and handle problems and analyse and evaluate different technical solutions from a holistic perspective	to demonstrate and discuss the software engineering theory with different groups (4)	Personal and professional skills and attribute; Interpersonal skills: teamwork and communication
show an ability to plan and, applying adequate methods, carry out assignments within given timeframes	to work in group and independent (5)	Interpersonal skills: teamwork and communication

¹² (*1) Programme Syllabus. Link:

<http://www.hkr.se/en/education/programme/?pCode=tbcsh&view=Plan>

¹³ (*2) Course Syllabus DA216A. Link: <http://www.hkr.se/en/education/course-page/?cCode=DA216A&view=Plan>

¹⁴ (*3) CDIO Syllabus. Link: http://www.cdio.org/files/project/file/cdio_syllabus_v2.pdf



show an ability to use knowledge critically and systematically and also to model, simulate, predict and evaluate processes starting from relevant information	to use theoretical knowledge in software development (6)	Disciplinary knowledge and reasoning
show an ability to design and manage products, processes and systems taking into consideration people's preconditions and needs and society's objective of economically, socially and ecologically sustainable development	to present the results in concretely and correctly way (7)	Interpersonal skills: teamwork and communication; Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
show an ability for team work and cooperation in groups of different composition, and		
show an ability to communicate orally and in writing and to discuss information, problems and solutions in a dialogue with different groups in English		
Judgement and approach	Judgment and approach	
show an ability to make assessments paying heed to relevant scientific, social and ethical aspects	to evaluate, on a profound level, ethical formulation of a question in main area of the education (8)	Personal and professional skills and attributes
be able to evaluate the possibilities and limitations of technology, its role in society and people's responsibility for its use, including social and economic aspects as well as environmental and work environment aspects, and	to evaluate the importance of the project work and working in a teamwork (9)	Interpersonal skills: teamwork and communication
show an ability to identify his/her need for additional knowledge and to continually develop his/her competence	to demonstrate, in a scientific way, deeper knowledge in some part of software engineering (10)	Personal and professional skills and attributes
	to discuss, in scientific and critical way, about the phenomenon, formulation of a question and different situations that exist (11)	Personal and professional skills and attributes

4.2 Software Engineering 2 DA540C – 15 ECTS

This course is given on the third year, first term of the Bachelor Programme in Software Development and Engineering.



How to increase the students' degree of involvement and participation in project based courses?

The difference between Software Engineering and Software Engineering 2 is the progression in student's critical thinking, and obtained deeper understanding of the main area. Also, as discussed earlier, CDIO and Demola are important educational and pedagogical parts in this course, with one large-scaled project.

The general entry requirements for this course are: Software Engineering 15 ECTS, and Digital Electronic System Design 7.5 ECTS or equivalent. Examination of the course is done through four examinatory parts:

- **Part 1** (5 hp) Software Engineering. Examination of expected learning outcomes 1-9 and 12. It is a group work, but examines individually. Examination grade is based on the mandatory attending on seminars, final result of the project, final project documentation, and oral presentation of the project. The grade will be given according to the scale U (fail), 3, 4, 5 where 5 is the highest grade.
- **Part 2** (2 hp) Laborations. Examination of expected learning outcomes 1 examines individually, and is based on laborations and evaluation of the corresponding tasks. Grades U (fail) and G (Pass) are applied.
- **Part 3** (6 hp) Project followup. Examination of expected learning outcomes 1-3 and 5-12 examines individually, and is based on examinations of project documentation on mandatory project meetings and related discourse. Grades U, 3, 4, and 5 are applied.
- **Part 4** (2 hp) Design Pattern. Examination of expected learning outcomes 4 and 12 examines individually, based on written report, and individual presentations of design pattern. Grades U and G are applied.

Further, we consider the following parameters: **1)** the programme syllabus with its intended learning outcomes (Kristianstad University, The Education Board for Health and Society, 2013); **2)** the Software Engineering 2, DA540C course syllabus (Kristianstad University, The Education Board for Health and Society, 2015); **3)** CDIO syllabus (Crawley, Lucas, Malmqvist, & Brodeur, 2011). Considering these, we have matched the parameters against each other, as it follows:

Software Engineering 2, DA540C matched against the programme syllabus based on four main areas:



- Scientific basis and software development
- Team work
- Planning and time frames
- Discussion with other groups and ethical aspects

Software Engineering 2, DA540C matched against the CDIO syllabus:

- Disciplinary knowledge and reasoning: 1- 4
- Personal and professional skills and attributes: 1-11
- Interpersonal skills: teamwork and communication: 3, 9, 11, 12
- Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context: 1- 12

Table 2 illustrates the three parameters matched against each other.

Table 2. Software Engineering DA540C matched to Degree learning outcomes and CDIO syllabus

Degree learning outcomes ¹⁵	Course intended learning outcomes ¹⁶ (*2) matched to Degree learning outcomes and CDIO ¹⁷ (*3) syllabus	
Knowledge and understanding	Knowledge and understanding	
show knowledge of the scientific basis of computer technology and its proven know-how and familiarity with topical research and development work, and	to demonstrate understanding about theories and techniques used when working in a software development project (1)	<ul style="list-style-type: none"> • Disciplinary knowledge and reasoning Personal and professional skills and attributes; • Personal and professional skills and attributes; • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
show broad proficiency in computer technology and relevant knowledge of mathematics and natural sciences and an ability to communicate this in English	to demonstrate understanding about different requirements categories, and how to recognize and describe requirements of the project (2)	<ul style="list-style-type: none"> • Disciplinary knowledge and reasoning; • Personal and professional skills and attributes; • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
	to demonstrate understanding of im-	<ul style="list-style-type: none"> • Disciplinary knowledge and reasoning; • Personal and professional skills and attributes;

¹⁵ (*1) Programme Syllabus. Link:

<http://www.hkr.se/en/education/programme/?pCode=tbcsh&view=Plan>

¹⁶ (*2) Course Syllabus DA540C. Link:

<http://www.hkr.se/sv/utbildningar/kurssida/?cCode=DA540C&view=Plan&version=9061>

¹⁷ (*3) CDIO Syllabus. Link: http://www.cdio.org/files/project/file/cdio_syllabus_v2.pdf



How to increase the students' degree of involvement and participation in project based courses?

	portance for planning and performing project work, regardless of what development part student do in the project (3)	<ul style="list-style-type: none"> • Interpersonal skills: teamwork and communication • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
	to demonstrate understanding about design patterns, and how to use those in the project (4)	<ul style="list-style-type: none"> • Disciplinary knowledge and reasoning; • Personal and professional skills and attributes; • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
Skills and abilities	Skills and abilities	
show an ability to independently and creatively identify, formulate and handle problems and analyze and evaluate different technical solutions from a holistic perspective	to apply different theories and techniques on software development project (5)	<ul style="list-style-type: none"> • Personal and professional skills and attribute; • Interpersonal skills: teamwork and communication • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
show an ability to plan and, applying adequate methods, carry out assignments within given timeframes	to carry out requirements list (6)	<ul style="list-style-type: none"> • Personal and professional skills and attribute; • Interpersonal skills: teamwork and communication • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
show an ability to use knowledge critically and systematically and also to model, simulate, predict and evaluate processes starting from relevant information	to apply design patterns when designing software (7)	<ul style="list-style-type: none"> • Personal and professional skills and attribute; • Disciplinary knowledge and reasoning • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
show an ability to design and manage products, processes and systems taking into consideration people's preconditions and needs and society's objective of economically, socially and ecologically sustainable development	to carry out software testing and write documentation (8)	<ul style="list-style-type: none"> • Personal and professional skills and attribute; • Interpersonal skills: teamwork and communication; • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
show an ability for team work and cooperation in groups of different composition, and		
show an ability to communicate orally and in writing and to discuss information, problems and solutions in a dialogue with different groups in English	to work self- dependent and in a group, and to do project presentation (9)	<ul style="list-style-type: none"> • Personal and professional skills and attribute; • Interpersonal skills: teamwork and communication • Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context



Judgement and approach	Judgement and approach	
show an ability to make assessments paying heed to relevant scientific, social and ethical aspects	to evaluate project management process to carry out a complete product, (10)	<ul style="list-style-type: none"> Personal and professional skills and attributes Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
be able to evaluate the possibilities and limitations of technology, its role in society and people's responsibility for its use, including social and economic aspects as well as environmental and work environment aspects, and	to show understanding of importance of project work and importance of cooperating with others (11)	<ul style="list-style-type: none"> Personal and professional skills and attribute; Interpersonal skills: teamwork and communication Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context
show an ability to identify his/her need for additional knowledge and to continually develop his/her competence	to demonstrate the ability to make assessment of software project work, informed by relevant scientific, social and ethical aspects, and also to demonstrate awareness of ethical aspects of research and development work (12)	<ul style="list-style-type: none"> Interpersonal skills: teamwork and communication Conceiving, designing, implementing and operating systems in the enterprise, societal and environmental context

The education in Software Engineering 2, DA540C is done through seminars, laborations, written report and artefacts, project meetings and presentations. The course contains 8 seminars. Every seminar is 3 hours long and is divided in the following parts:

- Teacher presents current scientific research in software engineering, during approximately one hour. Groups of 3-4 students are formed.
- Each group get one task to solve during approximately one hour.
- The rest of the time, approximately one hour, groups are presenting their solutions to each other in front of the class. In this way students reflect over presented solutions, and by training on skills in opposition technique.

Shaping seminars in this way, supports quality in learning (Kolb & Kolb, 2005) (Elmgren & Henriksson, 2013). More precisely the students' learning is facilitated through four learning phases:

- Students get concrete experience
- Students reflect and observe
- Students make abstract generalization



How to increase the students' degree of involvement and participation in project based courses?

- Students experiment actively

The main content in the course corresponds to eight seminars. These are presented in Table 3 below.



Table 3. Content overview in seminars (*S stands for seminar)

	Requirements and requirements analysis	Code of ethics Artefacts	User and System requirements Design patterns	Design UML	Software and Project Management	Introduction to a research project (Smart House)	Testing and Software Architecture	CMM CDIO Group Dynamics
S1	X							
S2		X						
S3			X					
S4				X				
S5					X			
S6						X		
S7							X	
S8								X

After taking the decision of replacing the lectures with the seminar teaching form, in Software Engineering 2, following course parts were connected to the *academic loop*: presentation seminar, opposition seminar, reflection seminar about the project work. Table 4 shows a sample of structure of seminar 1.



How to increase the students' degree of involvement and participation in project based courses?

Table 4. Example on content in seminars

	Reflection seminar	Presentation seminar	Opposition seminar
S1	Teacher present content of this seminar: understanding and analyzing requirements.	Group task is to imagine planning for starting up a company, where possible requirements shall be described and analyzed. This makes a good ground for further discussions between groups.	N/A

During seminars formative examination is applied. The use of formative examination in seminars makes it easier for teacher to understand student's learning style and to get possibility to influence student's learning in correct direction (Elmgren & Henriksson, 2013)

Two main changes were made in the course syllabus of the following examination parts:

Part 1, Software Engineering:

- Mandatory seminars are introduced and examined through active attendance, demonstrating critical thinking, making reflections and discussing regarding scientific research in the course's main parts.

Part 4, Design Pattern:

- Written report is introduced. Through the report students' abilities of searching and presenting scientific material, their critical thinking and reflections on the scientific content are examined. Also the report should be written on an academically level.5. Results

5.1 Evaluation using ZEFsurvey

The results presented next are based on evaluation using the ZEFsurvey, EvaSys and itslearning course platform evaluation forms. We will discuss further only the



evaluation made by using the ZEFsurvey in Software Engineering course, as it gives a clearer understanding on where the problems are and where we should act in order to facilitate students' learning.

Evaluator statistics			
Name	Evals	Answering	Answered
Software Engineering DA216A (HT14)	36	27	22
Total	36	27	22
Answering percentage	75%		
Finished	61.1%		
Interrupted	13.9%		
Did not participate	25%		

Figure 2. ZEFsurvey Evaluator Statistics

There were 40 registered on the course platform. Two were not active at all, and one left the course in the middle of it due to private reasons. So at the end of the course, there were 37 active students. Out of these 37 students, 36 chose to check the ZEF-evaluation, but only 27 started to answer it, and only 22 finished it. The figure below shows the result also in percentages.

We have then formulated a set of questions with regard to the course syllabus, as well as to the aspects covered in the CDIO-framework, such that survey is divided in 4 parts:

- Learning outcomes
- Working life relevancy
- Self-development
- General questions regarding the organization of the course.

We will discuss each of these, as it follows.

In the figure 3 below, learning outcomes are listed. The figure shows also their absolute values on how each of them were experienced by students with regard to the degree of importance to them. It can be noticed that the students were overall satisfied with the learning outcomes of the course.



How to increase the students' degree of involvement and participation in project based courses?

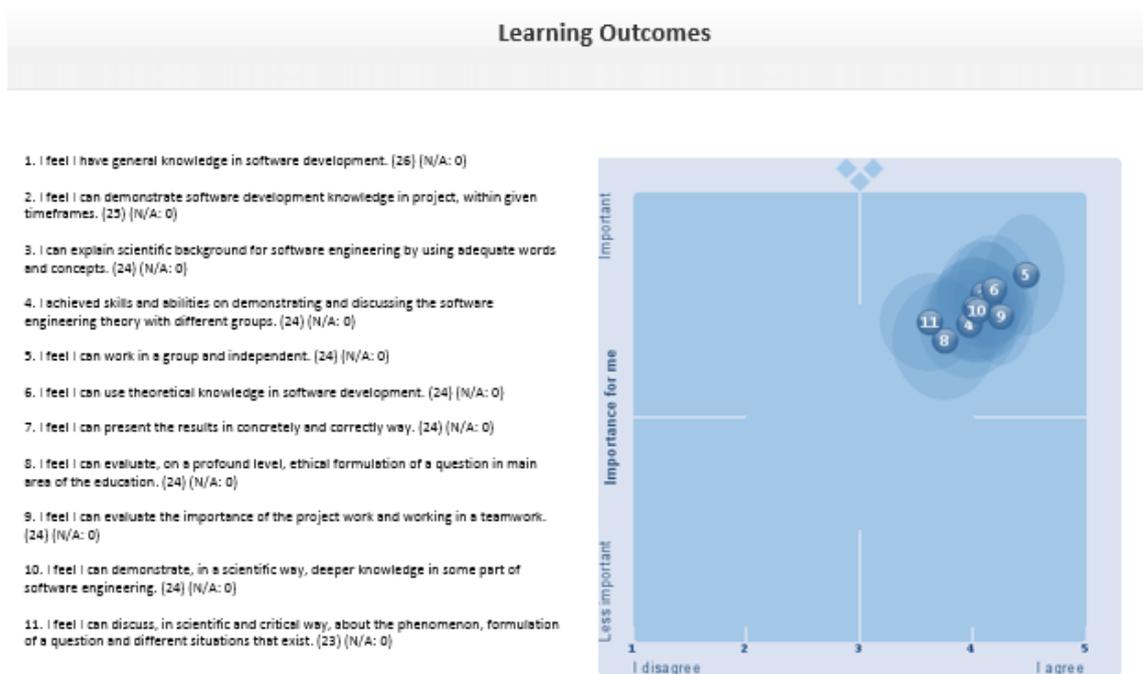


Figure 3. Absolute Values of the Learning Outcomes

We have then analyzed the relation between the absolute values of the learning outcomes by using the z-scoring method. The figure below shows the normalized distribution of the values, where the students showed most familiarity with the learning outcome 5, *I feel I can work in a group and independent*, whereas learning outcomes 8, *I feel I can evaluate, on a profound level, ethical formulation of a question in main area of education*, as well as learning outcome 11, *I feel I can discuss, in scientific and critical way, about the phenomenon, formulation of a question and different situations that exist*, should be reviewed and taken action onto.

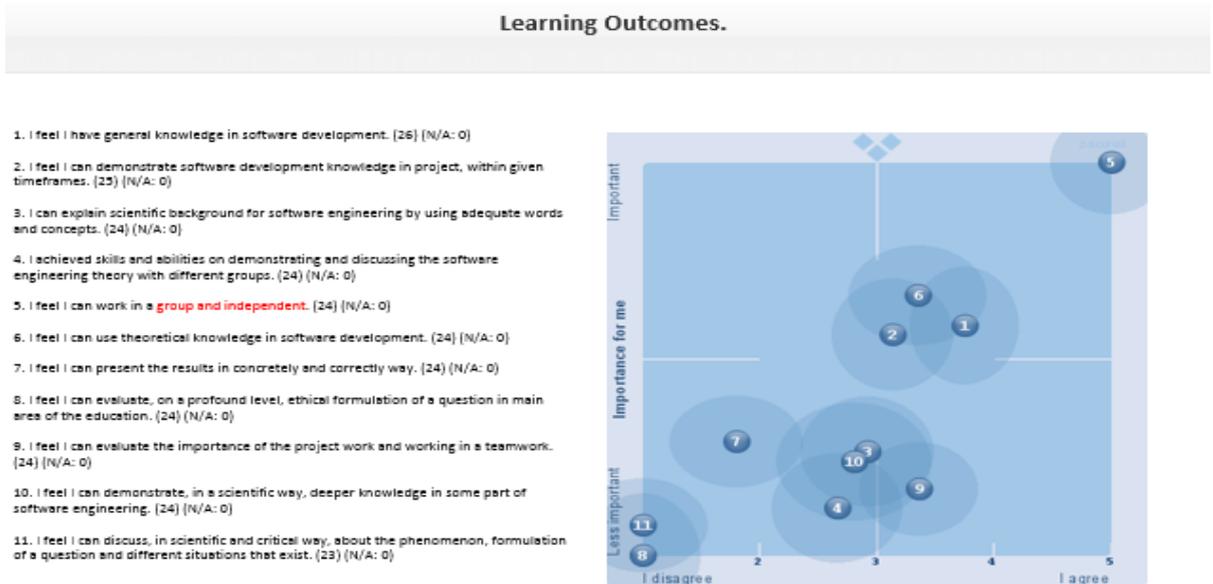


Figure 4. Normalized Learning Outcomes – Based on Z-scoring method

Further, we have asked a set of questions related to the working life relevancy of the course. The questions are stated below, whereas the results of the absolute, respectively normalized values, are shown in the figure X. Even here, the students had an overall well experience. However, the normalized values showed that we should take action on the fifth item, I had the opportunity to practice what I have learned during my studies.

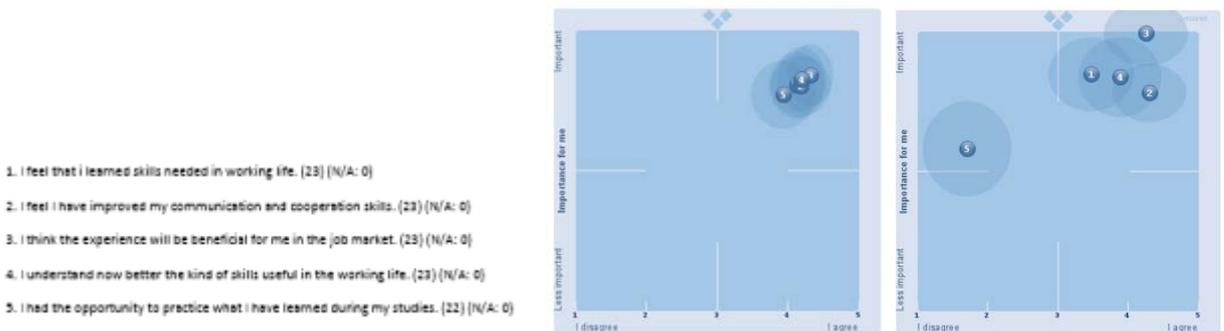


Figure 5-a. Set of questions

Figure 5-b. Absolute Values

Figure 5-c. Normalized Values

Figure 5. Working Life Relevancy



How to increase the students' degree of involvement and participation in project based courses?

1. The third set of questions contains only two questions, and these focused on self-development. Figure X below shows the absolute values, as well as the normalized values based on z-scoring.

1. I am motivated to work with real life problems. (22) (N/A: 0)
2. My skills matched with the requirement of the project. (22) (N/A: 0)

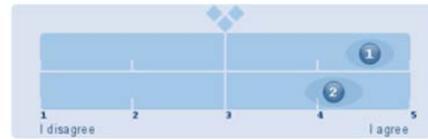


Figure 6-a. Set of Questions

Figure 6-b. Absolute Values

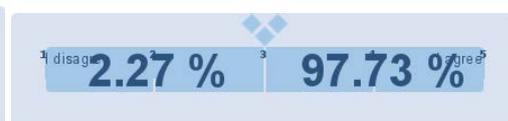
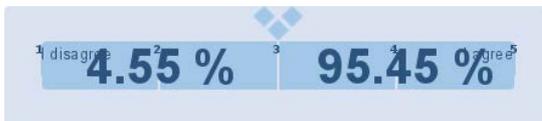


Figure 6-c. Normalized Value of the 1st Question

Figure 6-d. Normalized Value of the 2nd Question

Figure 6. Self-Development

Finally, the fourth set of questions were related to general questions on the course organization. Figure 7 below shows the questions, as well as the absolute values.

1. Project meetings (22) (N/A: 0)
2. Project management (22) (N/A: 0)
3. Online communication channel. (22) (N/A: 0)
4. Seminars (21) (N/A: 0)
5. Labs relevancy (21) (N/A: 0)
6. Communication (21) (N/A: 0)

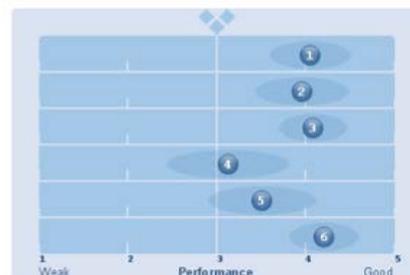


Figure 7-a. Questions Set

Figure 7-b. Absolute Results

Figure 7. General Questions

5.2 Outcomes

The attendance to the lectures given in the form of seminars has highly increased the attendance of students. They participated in the course in teams of 3-4 people. The final number of teams were 10. Forming the teams at the beginning of the course and allowing them to work on the seminars together, facilitated the interaction within the groups and between them. This contributed to an overall better evolution of group dynamics, such that the teams had time to mature in a longer



time span, and therefore also better results than previously were achieved. The outcomes were very positive in the form of innovative software projects with market potential (Sörensson, 2015). Here are a few to be named:

- Encryption and decryption algorithms for documents stored on the cloud storages
- Password generator application, such that the user no longer has to remember his or her passwords, but rather to use the application to store them in a secure way
- Sending of text between computer and mobile application
- Accessing all files stored on different cloud services through one application
- User identification through external devices, such as a smartphone

We observed that forming the student teams earlier in the process helped the teams to evolve faster in their group dynamics. The teams' evolution was mainly influenced by the seminars' structure, i.e. students had to prepare various assignment that were related to the corresponding disciplinary knowledge, while also reflecting on their own project. Moreover, the assessment was also easier to be made as the students' participation could be seen through a longer time span. Also, developing innovative software projects with market potential is a first proof of concept that seminar based teaching in the context of project based courses facilitate the project process. However, further studies need to be done, as our conclusions are based only on empirical observations. We will need to look further on other parameters as well, rather than only empirical observations, in order to see if the quality of teaching and learning is improved through introducing seminars based teaching in the context of project based course.

6. Discussion and suggestions for further work

This paper has discussed the structure of the courses in regard to: literature on the subject of *learning*; a set of changes were made in Software Engineering and Software Engineering 2 courses, as a part of the *Academic loop* (Swedish Higher Education Authority 2011-2014, Kristianstad University, 2015); governmental regulations, such as Swedish Higher Education Law, Swedish Higher Education Ordinance, Constructive Alignment and Bologna process; educational



How to increase the students' degree of involvement and participation in project based courses?

frameworks, such as CDIO, which was matched against the programme syllabus and courses syllabuses; changes that were introduced in the courses so far and their effect on the course through an evaluation made by using ZEFsurvey evaluation system; identification of the issues that need to be further researched and addressed, by using ZEFsurvey, such as: formulation of the research questions and discussions on the ethical aspects.

In the course of software engineering we will slightly modify the seminars structure, but we will still continue to teach the course in the form of seminars. Further, we will be evaluating the course Software Engineering 2 by using ZEFsurvey and we will implement the following changes:

- **Separate introduction and the seminars tasks.** During group presentations teachers observed student lacking in concentration, because of the size of the class group, and the long seminar duration. Thus, the first part of the seminars, where teachers present the content, will be moved outside the seminar. The first part is now named *Introduction to seminar*. It will be held as an own part before the seminar with group task.
- **Smaller seminar groups with two teachers.** Under group presentations, teachers also observed that the time was barely enough for examination. Due to this, more than two teachers will participate during presentations and discussions. Also presentations will be held with fewer groups attending the seminars.

Finally, we will continue to implement the seminars approaches in several of the project based courses, and in the future, also within the other programme, i.e. engineering programme, as a part of its own designed *academic loop*.



References

- (n.d.). Retrieved from www.cdio.org
- Adamson, L. (2013). *Kvalitetsutvärdering av högre utbildning - en kritisk granskning av det svenska systemet framtagen på uppdrag av SNS Utbildningskommission*. SNS Utbildningskommission.
- Altbach, G. P., Reisberg, L., & Rumbley, E. L. (2009). *Trends in Global Higher Education: Tracking an Academic Revolution - A report prepared for the UNESCO 2009 World Conference on Higher Education*. Paris: United Nations Educational, Scientific and Cultural Organization.
- Biggs, J., & Tang, C. (2007). *Teaching for Quality Learning at University*. New York, U.S.A.: Open University Press.
- Crawley, E. F., Lucas, A. W., Malmqvist, J., & Brodeur, D. R. (2011, June 20-23). The CDIO Syllabus v2.0 - An Updated Statement of Goals for Engineering Education. *Proceedings of the 7th International CDIO Conference*. (T. U. Denmark, Compiler) Copenhagen, Denmark: Technical University of Denmark. Retrieved 06 24, 2015, from http://www.cdio.org/files/project/file/cdio_syllabus_v2.pdf
- Demola. (2015). *Demola*. Retrieved 08 2015, 31, from www.demola.net
- Elmgren, M., & Henriksson, A.-S. (2010). Examination. In *Universitetspedagogik* (p. 245/325). Uppsala: Norstedts.
- Elmgren, M., & Henriksson, A.-S. (2010). Vad gynnar lärande? In *Universitetspedagogik* (pp. 49-60). Uppsala: Norstedts.
- Elmgren, M., & Henriksson, A.-S. (2013). *Universitetspedagogik* (2 ed.). Sweden: Studentlitteratur AB.
- Gjerde, S. (2013). *Coaching: vad varför hur* (2:2 ed.). Lund: Studentlitteratur AB.
- Granberg, O., & Ohlsson, J. (2009). Lärandets loopar. In *Från lärandets loopar till lärande organisationer* (3:1 ed., pp. 13-33/135). Lund: Studentlitteratur AB.
- Hedin, A. (2006). *Lärande på hög nivå - Idéer från studenter, lärare och pedagogisk*. Uppsala: Avdelningen för universitetspedagogisk utveckling, Uppsala Universitet.
- Högskoleförordningen 1993. (1993). Högskoleförordning (1993:100). Retrieved 08 13, 2015, from <https://lagen.nu/1993:100>



How to increase the students' degree of involvement and participation in project based courses?

- Kolb, Y. A., & Kolb, A. D. (2005). Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education. *Academy of Management Learning & Education*, 4(2), 193–212.
- Kristianstad University, The Education Board for Health and Society. (2013). *Bachelor Programme in Computer Science and Engineering, specialization in Embedded Systems*. Retrieved 06 24, 2015, from <http://www.hkr.se/en/education/programme/?pCode=tbcsh&view=Plan>
- Kristianstad University, The Education Board for Health and Society. (2015). *Software Engineering, DA216A*. Retrieved 06 24, 2015, from <http://www.hkr.se/en/education/course-page/?cCode=DA216A&view=Plan>
- lagen.nu. (1992). Högskolelag (1992:1434). lagen.nu. Retrieved 08 13, 2015, from <https://lagen.nu/1992:1434>
- Ljungblom, M., & Norberg, M. (2012). Inlärningsstilar. *En sammanställning kring de olika stilarna*. Visby: Högskolan på Gotland.
- SFS - Sveriges förenade studentkårer. (2013). *Studentens lärande i centrum - SVS om pedagogik i högskolan. Kvalitet ur ett studentperspektiv*. Stockholm: SFS - Sveriges förenade studentkårer.
- Swedish Higher Education Authority 2011-2014, Kristianstad University. (2015). *Computer Science, Evaluation Case: 411-52-14, 411-312-13*. Swedish Higher Education Authority 2011-2014.
- Swedish Higher Education Authority. (2015, 05 14). *www.uka.se*. Retrieved 06 23, 2015, from <http://kvalitet.uka.se/uppfoljningbristandekvalitet.4.197ecc1140ee238b5811.html?struts.portlet.action=/kvalitet/resultUppfoljningar&sv.url=12.197ecc1140ee238b581a>
- Säljö, R., & Södling, M. (2006). *Utbildning på vetenskaplig grund - röster från fältet - Tematiska studier*. Högskoleverket 2006, Utvärderingsavdelningen. Stockholm: Högskoleverket 2006.
- Sörensson, N. (2015, 05 30). Studenter presenterade nyttiga appar. Kristianstad: Kristianstadsbladet.
- Worldwide CDIO Initiative. (2015). *CDIO*. Retrieved 08 31, 2015, from www.cdio.org
- www.cdio.org. (n.d.).
- www.cdio.org. (n.d.).
- www.cdio.org. (n.d.).

