ADDRESSING INTERGRATED LEARNING THROUGH PROJECT-BASED COURSES – FIVE YEARS OF IMPROVEMENTS

PhD Daniel Einarson

Computer Science, Kristianstad University, Sweden

MSc Diana Saplacan

Computer Science, Kristianstad University, Sweden

ABSTRACT

Project-based educational forms are at the core of the CDIO concept, where students should be trained in contexts of complex enough tasks to prepare for the complexity of industry projects. Besides from fulfilling a project in itself, CDIO points out the importance of achieving integrated learning skills, including personal and interpersonal skills (CDIO Syllabus sections 2 and 3), where those are desired to meet the challenges of the working processes.

Projects in education moreover correspond to active learning, where students are encouraged to learn through solving the problems required to fulfill the goals of a project. Being active in the process of completing a project, does not only imply disciplinary training, but also training in achieving generic skills, such as experimentation, knowledge discovery, system thinking, teamwork, and communication. All in all, a conclusion is that student activities in project-based teaching and learning relate to all four sections of the CDIO Syllabus, and hence active learning will here contribute to integrated learning. Thus, activating students in project-based courses should have several positive values.

However, experiences show that one problem in project-based courses is that of activating a major part of a student group. Here, a common pattern is that some students are not contributing enough, resulting in other students covering up for them, or risking the whole project. Therefore, teaching efforts should be put on finding ways to widen the group of active students.

The project-based course Software Engineering 2, at Kristianstad University, Sweden, has undergone several years of improvements in order to, on one hand reduce the number of passive students, and on the other hand increase values of generic skills from the CDIO Syllabus. This paper will present development steps of that course. Methods, where some have been inspired from the Software Engineering industry, will be covered, and results of using those will be provided. A major result is that of increasing values of integrated learning, where this in itself contributes to the core of CDIO.

KEYWORDS

Project-based course, integrated learning experiences, CDIO learning outcomes, ethical aspects, capability-maturity model (CMM), self-awareness, Standards: 5, 7, 8.

INTRODUCTION

Project-based training is especially pointed out by the concept of CDIO, where values of project result is dependent, not only on the fulfilment of the project itself, but also on personal and

interpersonal skills. The compound value of students' performance is therefore an integration of abilities regarding all four sections of the CDIO Syllabus. That is, a project-based course is in itself a fundament for CDIO Standard 7, Integrated Learning Experiences. Furthermore, project-based courses are also examples on active learning forms (CDIO Standard 8), where learning activities concern all the four sections of CDIO Syllabus. That is, active learning has positive effect on achieving integrated learning skills. Project-based courses are valuable approaches to active learning, where a main effort for fulfilling the project lies on the students. The level of student activity may however be negatively influenced by aspects, such as, unclear working process models, and low grade of ambitions, which in turn have negative impact on integrated learning.

The project-based course, Software Engineering 2 (SE2), at the department of Computer Science, Kristianstad University, Sweden, has undergone several changes since 2010, when it started in the current shape. The students shall solve one main task (selected by teachers), where the level of scale and complexity of that task requires groups of about 15 students. The groups are then divided up into sub-groups of about 3 students, where sub-groups solve parts of the main task that eventually are integrated to fulfill the whole of the project. Teachers have here observed the above mentioned problems, and improvements to the course have been done to further develop students' activities as well as the integrated learning skills of those.

This contribution illuminates on how SE2 has undergone several steps of progressions, to meet and overcome the mentioned problems, with observed positive results. In short: At year 1 (2010) it was observed that students did not understand how to approach the work. To meet this, at the next year the, so called, Capability Maturity Model as a guidance of working was introduced. At year 2, it was observed that groups of students did not contribute enough, which led to the introduction year 3, of the Code of Ethics for software engineers. At year 3, it was still observed that some students were not contributing enough, which led to introduction of anonymous student evaluations. At year 4, the traditional lectures were replaced by student active seminars. Finally, at year 5, the seminars format was changed to activate students even further. Moreover, the anonymous evaluations were refined. Thereafter an additional step of repetitive evaluations of generic skills has been introduced, to push the ambition on integrated learning skills at an even higher level.

The aim of this contribution is to give a complete view on progressions made, and to serve as a possible guidance in circumstances of implementing courses such as SE2. Next section will give a more detailed view on the course. Thereafter, course improvement steps will be covered, and conclusions will be provided.

A BRIEF PROJECT OVERVIEW

The course *Software Engineering 2* has in its current shape been provided since 2010, where the structure of the project is intended to be complex enough to require up to about 15 students to fulfill the project. The student group as a whole is here divided up into sub-groups of about 3 students per group, putting significant demands on collaboration and communication. Meanwhile, the project task should be technically complex enough to address challenges outside the core of the rest of the courses' contents, with the implication that students have to extend their disciplinary skills through own discovery initiatives.

Project Structure

The core of the project task is to develop a system with smart house techniques, where the system is to be used by people with functional disabilities. There have been other project tasks

too, but for reasons of simplifications, a focus is put on smart house projects. The system interacts with devices, such as lamps, temperature sensors, and radiators. Moreover, the system includes a server, database, and parts for communication. At the front end part, apps at smart phones, and web-based interfaces are used for human-system communication. Figure 1 illustrates this, as also illustrated in (Einarson, 2011).

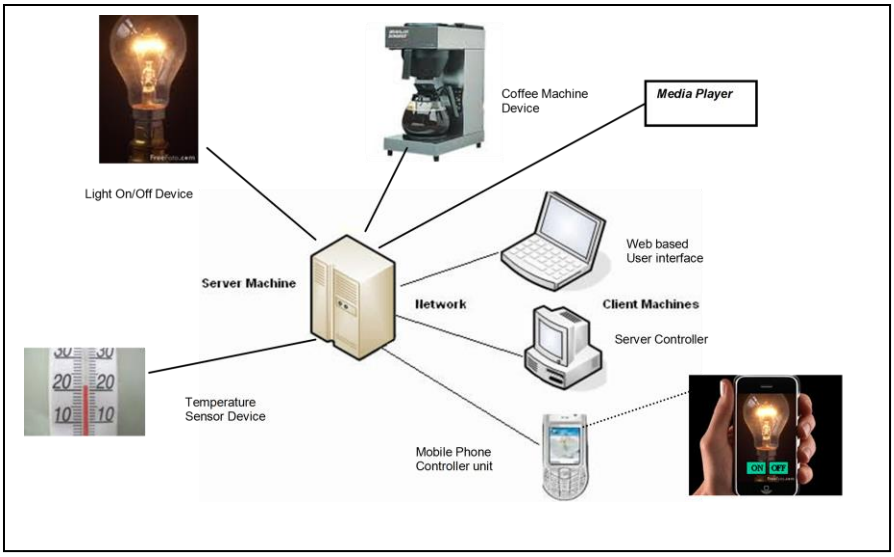

Figure 1. System overview of student project.

To be able to fulfill the project as a whole, the student group has to be divided into sub-groups, where those develop their specific part of the system, or project. Typically, as show in Figure 2, (Einarson, 2011), a part may correspond to functionality for devices of the smart house, middleware, or interactive human user controller parts. The group as a whole has a project leader, and a requirements manager. Moreover, each sub-group has one specific sub-group leader.

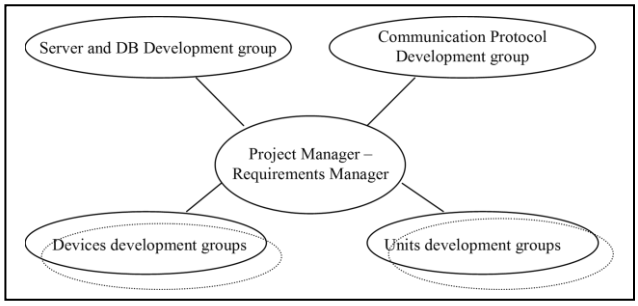

Figure 2. Project group structure.

The course is on 15 academic credits, and runs over one semester. The structure of the working process includes several iterative meetings with teachers, where the project is discussed so far. Before those meetings the groups shall upload a number of artefacts that should clearly describe aspects, such as, design, project risks, requirements, supplementary requirements, and more.

The Project's Correspondence to CDIO

It can clearly be seen that the project corresponds well to all four sections of the CDIO Syllabus. Therefore, CDIO Standard 7, *Integrated learning experiences*, should be addressed (still, without involvement of stakeholders outside education), and also Standard 8, *Active learning*. The project does furthermore clearly correspond to several points of level 2 of CDIO Syllabus (CDIO Syllabus). For instance, point 2.2 is required since *experimentation, investigation and knowledge discovery* all are needed to approach, and fulfill the project. 2.3 *System thinking* is also clearly addressed. 3.1 *Teamwork* is at the core of the project, as well as 3.2 *Communication*, and international students do furthermore contribute to 3.3 *Communication in foreign language*. Points 4.3, to 4.5, are all obvious (please see more on (CDIO Syllabus) for more on the details), and 4.1 *External, Societal, and Environmental Context*, is addressed from the point of the problem domain, that is, people with functional disabilities. Moreover, seen from perspective of CDIO Syllabus, level 3, there is further correspondence to, e.g., 2.4.1 *Initiative and the Willingness to Make Decisions in the Face of Uncertainty*, 2.4.3 *Creative Thinking*, and 2.4.7 *Time and Resource Management*. Those points are motivated by the fact that the strategies to solve the project should be left to the students themselves. This is a fact that also contributes to finding strategies to meet point 2.4.6 *Lifelong Learning and Educating*, which closely relates to points 2.4.1, and 2.4.3.

PROGRESSIONS – FIVE YEARS OF IMPROVEMENTS

For purposes of motivating the project, and preparing for future of students, structures of the course have been inspired by methods and principles from the Software Engineering field as it is established in industry. This relates to the working process at large, as well as on levels of personal attitudes.

First Year

The style of the working process was set. An iterative and incremental style was chosen, that lays close to the Rational Unified Process, or rather UPEDU, the Unified Process for Education (UPEDU, 2014), but simplified to exclude much of the extensive work on documents. Please see in (Einarson, 2011) for more on that, and how it is furthermore explained how well the Software Engineering field may influence CDIO-based educational forms.

Even though several of the CDIO Syllabus Learning Outcomes seemed to be met by the course, the teacher (one of the co-authors of this paper) found it hard to see the value of the course, with respect to student results. It seemed to be hard to see the actual individual contributions, and the teacher's assumption was that parts of the student group did not contribute enough. The course evaluation showed that students understood the goal of the project as such, but did not understand the process of getting there, they wanted the teacher to be more clear on that and provide guidance in that direction during the project. That was also the main point of criticism that the teacher carried to the next year the course was provided.

Second Year

The teacher's conclusion from the first year was that it would be of higher values if students decided about their process more by themselves than if the teacher should do it. To meet this, the concept Capability Maturity Model (CMM), from Software Engineering (Sommerville, 2010), was clearly introduced. The CMM can be explained as a five-level model for development organizations, where each next level corresponds to a higher level of organization maturity. Figure 3, illustrates CMM.

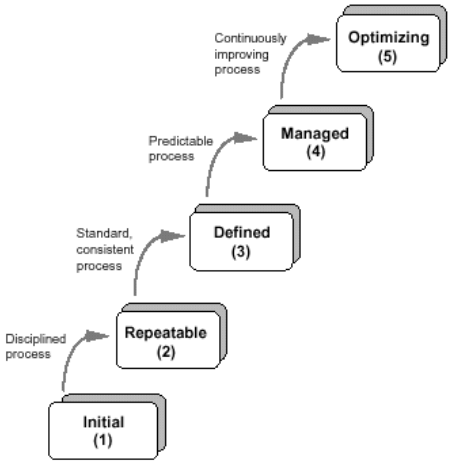

Figure 3. Levels of the Capability Maturity Model (CMM).

Level 1, corresponds to an ad hoc way of working, which also often is a way that student projects are done. Level 2, corresponds to a defined way of working, typically set by a project manager, and where the development team has to follow this. In the context of the project-based course, SE2, the first year corresponds much to that level. The teacher here instead wanted to simulate level 3 at the course. This means that the process should be defined by the students themselves, resulting in much more effort on learning outcomes, such as, teamwork and communication, attitudes thought and learning, and others.

The result, as presented in (Einarson, 2012), shows that the course through this approach undergone great improvements. Results were clearly better, both when it came to the final project result, and the process of getting there. It could also be shown that values of several learning outcomes were improved.

However, during the process it was clearly shown that sub-groups of students did not contribute, with frustration at student project managers, and risks for project fulfilment, as results. The teacher tried to meet such problems with arguments, such as, ethical behaviour and moral, but had hard to get through. While the course result was generally good, that was the main problem the teacher carried on until next year.

Third Year

In order to reduce the number of passive students, and to meet the challenges of being active, ethical aspects were introduced. Especially, the *Code of Ethics* (CoE) for Software Engineers was especially emphasized (Gotterbarn, 1999), where students should reflect on this concept, and put those reflections in the context of the project. The CoE, as presented in the table below, constitutes a list of several aspects, such as, on software products, and development of software, and is a result of a co-operation between the ACM, and IEEE-CS.

1. **PUBLIC** - Software engineers shall act consistently with the public interest.
2. **CLIENT AND EMPLOYER** - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. **PRODUCT** - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. **JUDGMENT** - Software engineers shall maintain integrity and independence in their professional judgment.

5. **MANAGEMENT** - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. **PROFESSION** - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. **COLLEAGUES** - Software engineers shall be fair to and supportive of their colleagues.
8. **SELF** - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

While all of the points were under discussion, especially point 7 was emphasized. By addressing this, it was clarified that this was not only for the purposes of being fair to co-students, but actually a phase of preparing for future employability, and work, through an accepted way of looking at the profession. In (Einarson, 2013), it is shown that the introduction of the CoE at the course contributed with a correlation between how seriously students approached the CoE and the project result. Besides from those inspirational results, another implication is that further CDIO Syllabus Learning Outcomes, such as, 2.5.1 *Ethics, Integrity and Social Responsibility*, 2.5.2 *Professional Behavior*, and 2.5.6 *Trust and Loyalty*, are addressed to some degree, as well as 3.2.8 *Negotiation, Compromise and Conflict Resolution*, 4.1.1 *Roles and Responsibility of Engineers*, and 4.1.2 *The Impact of Engineering on Society and the Environment*.

What could clearly be seen was that the number of active students was increased. Still, as a common problem in project-based courses, some students were relying on the loyalty of others, where those were covering up for them. To increase the possibilities for the teacher to catch information on such occurrences, anonymous student evaluations to find the state of the course, and especially co-student contributions, was taken under consideration.

Fourth Year

During the fourth year, the course content was kept the same, but we introduced anonymous evaluations in paper format conducted during each project meeting, besides the formal digital anonymous evaluations conducted at the end of the course. The aim was to give the students the opportunity to raise their voice regarding the work and colleagues. The course instructors would keep those and go through those, both after each formal project meeting, in order to keep track of the project progress, and team evolution. The instructors would go through those both during the course, and at the end of it. The purpose was to reveal and counteract eventual destructive loyalty where students were covering up for others (as identified during the third year). In this way, the students were given the chance to raise their voice, while still remaining anonymous, and the information revealed remaining confidential. The format was on paper, i.e. non-digitalized format, and was based on informed consent. The evaluations were done on a voluntarily regularly basis, i.e. at every project meeting. Moreover, this also helped the students, on a meta-level, to rate their own contribution in regard to others, and at the same time relate to ethical aspects of CoE (see 7.Colleagues and 8.Self earlier discussed under Third Year section). In a way, this was calibrating their attitude towards themselves in the first place, towards the teachers, towards colleagues, and the project in itself. Furthermore, the anonymous evaluations could present foundation of eventual ongoing conflicts. This, in itself, increased the value of CDIO Syllabus Learning Outcome 2.5.2 *Professional Behavior*, 2.5.6 *Trust and Loyalty*. 3.2.8 *Negotiation, Compromise and Conflict Resolution* and 2.5.1 Ethics, Integrity and Social Responsibility, more specific, *The moral courage to act on principle despite adversity* and *The possibility of conflict between professionally ethical imperatives,* and *Truthfulness*. In addition, in order to encourage students' active learning, the teachers decided to proceed with changing the format of the course, such that seminars were introduced during the next year.

Fifth Year

(Hedin 2006) distinguishes between two ways of teaching: "*Sage on the Stage*" and "*Guide by the Side*". Whilst the former one refers to the more traditional way of teaching, i.e. through lectures where the teacher takes an active position during lectures, the latter one refers to teaching forms that are more focused on the active involvement of the student, the teacher becoming more of a facilitator. At the same time, a UNESCO report since 2009, supported the idea that *"lecturing about declarative knowledge can no longer be the default method",* as shown in (Saplacan et. al. 2015). In this way, during the fourth year, the way the course was provided was shifted from traditional lectures towards a more "*Guide by the Side*" approach, as in (Hedin 2006). The main teaching form introduced here was a hybrid form of seminars. Initially, the seminars were structured into three parts: first, introducing a topic (that previously was a lecture); second, the students would get hands-on group assignments to work on that were relevant to the topic presented in the first part of the seminar; and finally the groups had to present their summary of discussions for each of the assignments. Here, the groups were the same as the sub-groups from the project. The intention of this way of approaching seminars was multifaceted: (1) on one hand, to create an additional discussion platform for the sub-groups which would eventually enhance the cohesion of the sub-groups; (2) to enhance the communication and cohesion between sub-groups, in order to prepare them for the later phase of the project, where all sub-groups should integrate their work, hence the cohesion of the team as a whole; (3) to discover eventual conflicts within the sub-groups that would not be acknowledged during formal project meetings; (4) to provide additional opportunities for oral presentations; and (5) to examine new perspectives on the project, but also outside of it; (6) to train skills such as time-management at a smaller scale, as the assignments given had a limited amount of time to be solved and presented. This would cover at some degree CDIO Syllabus Learning Outcomes 3.1.2 *Team Operation*, 3.1.3 *Team Growth and Evolution*, 3.1.4 *Team leadership*, 3.2 *Communication*, 3.2.6. *Oral Presentation*; 3.2.7 *Inquiry, Listening and Dialog*, 3.2.8. *Negotiation, Compromise and Conflict Resolution*, 3.2.10 *Establishing Diverse Connections and Networking*.

Although the intentions were good, the teachers (one of the co-authors of this paper) experienced that the format was a compressed one and it would be better to separate the introductory lectures from seminars. This would give the students more room for preparations and the discussions would become deeper.

CURRENT AND FUTURE APPROACHES

During the next years, the content was kept the same. Though, the main focus here was on *optimizing* the process. Hence, the format of the seminars was changed from the hybrid version of one block with three parts (introductory lecture, assignment and presentation), into two main blocks. First, an introductory lecture was given, whereas the students were assigned a set of around 15-25 questions per seminar, as a preparation. Some of the questions, or variations of those, were chosen as topics for the seminar. The students had the opportunity to discuss first in group, and then present their views or reflections, as well as having the opportunity to argue for- and against for the choices made. The main observation here was that the discussions were more dynamic and interactive. In addition to the CDIO Syllabus Learning Outcomes covered during fourth year, here were observed tentative of 3.2.7 *Inquiry, Listening and Dialog* and 3.2.9 *Advocacy* as well. In addition, regarding the anonymous evaluations, experiments on two dimensional self-evaluations have been introduced. We discuss this under the next section.

EXPERIMENTS ON SELF-AWARENESS

While approaches presented above have shown how values of learning outcomes of project-based courses may increase, questions arise on how to increase those even further. To address this, experiments have been done, as outlined in (Einarson et al, 2016), where students should themselves be aware of the educational meta-process they are undergoing. Here students are exposed to iterative self-evaluations, where they respond to statements based on learning outcomes, such as those pointed out in this contribution. For that self-evaluation an online evaluation tool is used, which in turn provides a base for further discussions within the student group. While this approach still is at an experimental level, it is of significant interest to study possible impact on students' project work, as well as on their level of self-awareness. A corresponding CDIO learning outcome of this task, may here be 2.4.5 *Self-awareness, Metacognition and Knowledge Integration*.

To further elaborate on this, while during previous years the main focus was on evaluating own' and colleague's work, this time we have introduced two dimensional (2D) self-evaluations using ZEFsurvey tool. Please see for more on (2D) self-evaluations the work in (Saplacan and Teljega 2015; Saplacan, Silvén, and Einarson 2016; Einarson, Saplacan, and Silven 2016; Einarson and Saplacan 2016). The statements (compare to *questions*) in the (2D) self-evaluations addressed: course learning outcomes, working life relevancy of the course/project, self-development and general questions regarding the organizations of the course. The statements were formulated by the course instructors using the pronoun "I", e.g. "I feel I learned skills needed in working life", "I feel I have improved my communication and cooperation skills", "I think the experience will be beneficial for me in the job market" etc. The students had to place their answer on a 2D-scale (compare to surveys with one scale), where one axis represented the importance of the statement of them, and the other axis represented if they degree at which they agreed. In this way, by using the pronoun *I* and the 2D-scale, the students would become the main subject of the statements, and at least if the answers were not very exact, they would at least encourage self-awareness on how the student is positioning him/her self towards the project/course. This contributed to both students' and teachers' self-awareness. We advance our discussion around this topic in the sequel.

Elements of self-awareness have been developed even on the teachers' side. For instance, if we take the capability-maturity model (CMM), we have noticed that during the years, from a pedagogical perspective, we have taken the same steps as in CMM. Whilst in the initial phase (1), we have developed ourselves an understanding of working with CDIO-oriented projects, during the repeatable phase (2), we have introduced the CMM model. Further, during the phase (3), the *defining* phase of CMM, we have framed CoE for the students, but it also helped us in developing deeper understanding in the way we anchored the project – for the students. Moreover, we have introduced the anonymous evaluations, as a *tool* for conflict resolution and support for CoE. During the fourth phase (4), we have worked with the course from a *"management"* point of view, by keeping the same content, but re-structuring its format, through a hybrid version of seminars. Lastly, during the last phase, (5) *Optimization*, we worked with the optimization of the course, i.e. for instance, keeping the same content, but improving the format of the seminars, and by introducing 2D self-evaluations, which would contribute in their turn to both students- and teachers' self-awareness. For instance, based on the results of the surveys, the teachers would eventually change the format or content of the course, course syllabus or learning outcomes. Other forms of changes that were applied was eventually to modify courses that were mandatory entry requirements for this course. Although our arguments here may not be described in-depth, we can see the correspondence between

CMM and our pedagogical approach. At the same time, we intend to develop this approach further, in our future work, but for now we wanted to introduce it briefly in order to get a basic understanding of the concept, as an element of self-awareness on teachers' side. We call this approach the CMM Learning Staircase. This aims also to be a potential guideline for those who would like to learn from our experience and aim to introduce CDIO-oriented projects.

However, learning is a cyclic process as show by Kolb in (Ljungblom et. al. 2012) (Jansson and Ljung 2015). So once we reach stage (5), Optimization, a new iteration through all cycles should be taken, at deeper level. For instance, by introducing different ethical frameworks that can be applied not only towards self- and colleagues, or teamwork, but also regarding the socio-technical context, see, for instance, the ethical frameworks presented in (Ess 2010).

SUMMARY

This contribution has covered several approaches to increase levels of activation of students, through several years of experiences of a project-based course in Software Engineering. Moreover, activation has been specified around a number of CDIO Syllabus learning outcomes. Those learning outcomes have been shown to reflect several levels of all four sections of CDIO Syllabus, clearly motivating that the course corresponds well to CDIO Standard 7, on Integrated Learning Experiences.

The course has, under the years, being well established, even though further experiments to further increase values of learning outcomes are implemented. While the specific project task, described in this contribution, may vary, a fundament of that establishment is still valid, and may be used generically in other project-based courses, in Software Engineering, but certainly also in courses of other disciplines.

REFERENCES

ACM. Association for Computing Machinery, Advancing Computing as a Science & Profession, "Software Engineering Code of Ethics and Professional Practice", http://www.acm.org/about/se-code

CDIO Syllabus. www.cdio.org, Knowledge Library, http://www.cdio.org/framework-benefits/cdio-syllabus

CMM. On Capability Maturity Model. http://ibiblio.org/gferg/ldp/SCM-OpenSource/scm-traditional.html

Einarson D. (2011). Working- vs. Educational Processes in Software Engineering vs. CDIO, *7th International CDIO Conference*.

Einarson D. (2012). Approaching CMM to an Educational CDIO based Software Engineering Process, *8th International CDIO Conference*.

Einarson D. (2013). Ethics and Responsibilities in a CDIO based Software Engineering Process, *9th International CDIO Conference*.

Einarson D., Saplacan D. & Silvén P. (2016). Approaching Work Integrated Learning through Learning Outcomes and Evaluations, *12th International CDIO Conference*.

Einarson, D. & Saplacan, D. (2016). A proposal of learning outcomes for work integrated learning based on 2D evaluation methods, *Lärarlärdom 2016, Kristianstad, Sweden, publish in progress*

Ess, C. (2010) Digital Media Ethics, Ch. 6., *MA Policy Press, Cambridge*.

Gotterbarn, D. (1999). How the new Software Engineering Code of Ethics affects you, *IEEE Software 16*.

Hedin, A. (2006). Kapitel 10 - Vad Främjar Lärandet? I Boken Lärande På Hög Nivå - Idéer Från Studenter, Lärare Och Pedagogisk Forskning Som Stöd För Utveckling Av Universitetsundervisning. *Uppsala Universitet, Avdelningen för universditetspedagogisk utveckling, PU tidigare UPI.* http://www.uadm.uu.se/upi/arkiv/rapporter/Larandepahog%20niva.pdf.

Jansson, T. & Ljung, L. (2015). Individer, grupper och ledarskap i projekt. 1:4. *Lund, Sweden: Studentlitteratur AB, ISBN 97891440680077*.

Ljungblom, M. och Norberg, M. (2012). En sammanställning kring de olika lärstilarna - Inlärningstilar. Visby, *Högskolan på Gotland.*

Saplacan, D., Silvén, P., & Einarson, D. (2016). CDIO - A Convergence Point for Academia and Companies, work in progress paper presentation, Turku/Åbo, Finland, *12th International CDIO Conference*.

Saplacan, D., & Teljega, M. (2015). How to increase the students' degree of involvment and participation in project based courses? *Lärarlärdom 2016*, *Växjö, Swden, Linnéuniversitetet edition*.

Sommerville I. (2010), Software Engineering, Ninth edition, *Addison Wesley, ISBN-13: 978-0137035151 , ISBN-10: 0137035152.*

UPEDU (2014), Rational Unified Process in Education, UPEDU, http://www.upedu.org/

BIOGRAPHICAL INFORMATION

Daniel Einarson has a PhD in Computer Science and has several years of experience in teaching Computer Science and Software Engineering. Furthermore, he has been experimenting with several different forms for project-based learning.

Diana Saplacan has an MSc in Embedded Systems and a BSc in Computer Software Development from Kristianstad University, Sweden. She has also been involved in interdisciplinary projects, participatory (action) research, pedagogical projects, and has teaching experience in Software Engineering courses. She is currently a PhD candidate at University of Oslo, Norway.

Corresponding author
Dr. Daniel Einarson
Kristianstad University
Elmetorpsvägen 15
291 88 Kristianstad
Sweden
+46 -44 203177
daniel.einarson@hkr.se