When citing, please refer as:
Heikkinen, Kari & Porras, Jari & Ikonen, Jouni & Hämäläinen, Harri: Emphasizing Group Work In Code Camp Process, Journal of Education, Informatics and Cybernetics, 2009, vol. Vol.1, nro. 2, p. 25-31, ISSN 1943-7978

Emphasizing Group Work in Code Camp Process

Kari HEIKKINEN, Jari PORRAS, Harri HÄMÄLÄINEN and Jouni IKONEN
Department of Information Technology, Lappeenranta University of Technology
Lappeenranta 53851, Finland

ABSTRACT

Code camp is a teaching method in which people learning programming study together intensively for a short period of time. Code camps are designed to promote collaborative learning of necessary programming skills. In this paper, the code camp process is studied; especially the part of the process that is built on the characteristics of group dynamics. The code camp process is evaluated through one of the code camps realized at Lappeenranta University of Technology in 2008. The evaluation reveals that a code camp setting is a promising teaching method which supports collaborative learning.

Keywords: Code camp, Collaborative learning, Group dynamics, Programming, Process

1. INTRODUCTION

A computer science student is required to master a set of skills when he or she graduates from the university. Often the profession-oriented skills are the most important. Essentially, for a computer science student these skills include i) design and creativity, ii) implementation, i.e. programming and iii) evaluation and analysis. However, skills such as critical thinking, problem solving, information retrieval, collaboration and interaction are almost equally important.

Collaborative learning, as referred to in this paper, is seen as an approach to learning in which the students solve problems in a group and each student retains the same learning outcomes from the course. Preston [1] has presented five critical attributes for successful collaborative learning: i) common task or learning activity suitability, ii) small group learning, iii) cooperative learning, iv) interdependence and v) individual accountability and responsibility.

The code camp approach can be defined as a collaborative learning setting that aims to promote profession-oriented skills. In code camp, the term camp refers to a situation where students assemble and stay a while together. The term code refers to coding, i.e. writing computer programs. During a code camp, students write programs together, solve problems related to their work together, eat together and even might relax together in a sauna. Even though the primary objective of the code camp approach is to enable the learning of programming, learning design and creativity skills as well as evaluation and analysis skills is present in the learning outcomes. The time spent intensively together gives the opportunity to work on ideas and promotes the possibility (and the need) to interact with other people working in the same situation/place. By emphasizing the social aspects, e.g. the code camp spirit, learning can be done in a more meaningful way [2]. The grading system for code camps promotes both the characteristics of collaborative learning and the code camp spirit. Table 1 presents how Preston's attributes are visible in the code camp approach.

Table 1. Collaborative learning attributes within the code camp approach

Attribute	Realization of the attribute in the code camp approach
Common task or learning task suitability	Learning outcomes are similar for each and every participant; e.g. at the end of the course the student is able to demonstrate the problem solving by an application that fulfills the common task criterion/criteria
Small group learning	The group size has normally been from two (2) to three (3) persons
Cooperative learning	A better grade can be obtained by helping others
Interdependence	Learning outcomes and activity are fixed and thus positive interdependence can be reached
Individual accountability and responsibility	Students will take an individual test and will receive an individual grade

As can be seen, the code camp approach is very much in line with Preston's collaborative learning attributes. However, the process (of creating a new course) is time and resource consuming, as it requires a great deal of staff resources to design, run and evaluate. The current status of the code camp approach is as follows: common task and learning outcomes are quite straightforward to create and evaluate. Small group learning could be better utilized, whereas now the group dynamics are neglected. Cooperative learning could be enhanced by forcing co-operation, but we would prefer it to be more natural interaction. Interdependence is clearly addressed. Individual accountability is also addressed but should be more transparent.

The rest of the paper is structured so that the second chapter introduces the current code camp process with an emphasis on group dynamics. The third chapter analyses

the process. The final and fourth chapter concludes the paper and presents some future plans in the development of the code camp approach.

2. CODE CAMP PROCESS OVERVIEW

The code camp process is based on collaborative learning efforts in a very short time period. Thus the code camp is an ideal environment for group focused actions. This paper emphasizes group dynamics in the code camp process. The three building blocks of group dynamics are i) understanding of the group processes, ii) empowerment of the students and iii) evolvement of the group. Here, a group is seen as a small group that forms a social system which is built on the different elements of interactive students. Group dynamics, on the other hand, as understood within the code camp approach, can be defined as processes and skills that occur and shape during the life cycle of the group; the characteristics of these processes and skills are unique to the students within the group. The group processes should support the positive effect for skill development. Thus, it is essential that the process supports i) an awareness and understanding of the learnt skills, ii) the effects of the skills on different students, iii) the effects of those skills on the overall objectives / tasks of the group and iv) a guidebook for teachers, enabling different students to become active members of the group and helpful for the event itself.

According to Åberg [3], processes that compose group dynamics are *group spirit*, *group norms* and *group synergy*. In addition to that, the skill level of the students should be at an equal level, if possible. The activities within the group should be clearly defined and supported by the staff. It should be noted that group dynamics are not formed instantly. According to Lindblom-Ylänne and Nevgi [4], the group will normally go through the following five (5) phases; i) forming, ii) storming, iii) norming, iv) performing and v) adjourning. The code camp process should support the group forming as efficiently as possible.

Figure 1 gives an overview of the code camp process. It has excluded the design and evaluation phase, as the figure aims to illustrate the student viewpoint on the code camp process. The process applies Kovanen's [5] approach in which learning can be supported through four unique dimensions; i) context, ii) situation, iii) reflection and iv) processes for knowledge and skills. In the code camp process, the context is separated from the situation even though the two are cross-related. When analyzing human cognitive operation, context and situation can both mean physical set-ups for learning as well as emotional and social factors. This means that the situation and the context affect the knowledge obtained from the teaching event. This might be problematic, as information transfer to other situations might not occur. Thus, a distinction is made between context and situation. The situation is clearly the art of doing things in the fashion of code camp, and the actual learning outcomes are not mixed with the situation. The learning outcomes are context-specific. In addition to Kovanen's approach, time as another dimension was added because code camps are intensive events.

In general, in most courses, the context is the programming of communication engineering applications. The context defines the learning outcomes, effects on the situation (in terms of equipment, software, place, staff support), reflections as part of creating knowledge / skill and evaluation criteria (not visible in the figure). Some of the learning outcomes are purely context-dependant, e.g. measure a skill of programming of a selected programming language.

The students have to reflect continuously on the aspects learned. The tasks and check points have been inserted into the timeline (between context and process in Figure 1), and they are supposed to aid the learning. However, this is problematic, as the groups are often in different phases or have performed differently. The understanding of group dynamics and group phases would clearly help.

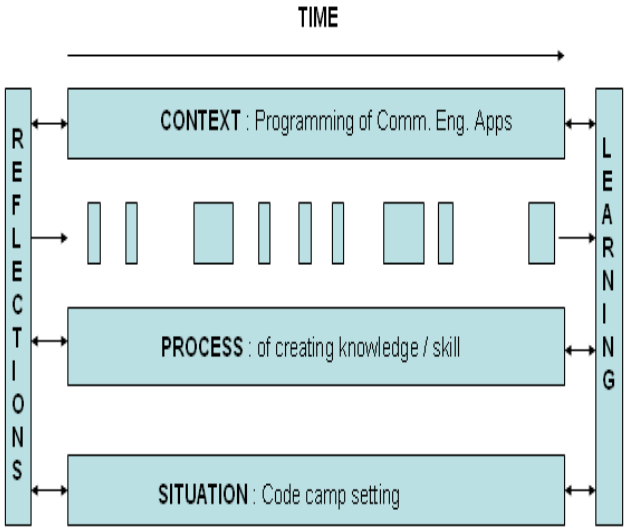

Figure 1. Code camp process overview

Related Work

Joseph and Payne have studied group dynamics and collaborative group performance [6]. They present well argued evidence of the benefits of group work. These benefits include analytic, communication and behavioral skills which are often omitted. They also point out that students become aware of the significance of small group dynamics as a tool for task achievement and success in a team environment. In their opinion, group dynamics is not

a simple matter. As their study had a different purpose than our code camp approach, the analysis results cannot be directly applied. However, their results indicate that the more students participate in collaborative group work, the greater are the students' achievements.

Dooner et al. [7] have analyzed the publication "The Social Psychology of Organizing" in order to find the social dynamics of a small group. Their purpose was to find how the groups really get started and stay at a sustainable level. They provide an excellent set of related work in the area of group work, e.g. on the nature of collaboration. The findings cannot be directly transferred into code camp, as the research process took years, which surely means that the intensity level of collaboration drops when the time is long.

Ya-Hui Chang provides information concerning the influence of group processes on the individual learner [8]. She points out that "while learning may seem to be an endeavor of an individual, most learning situations take place in groups". This clearly shows that the student is significantly affected by the group in which he or she works. She also cites Schmuck and Schmuck and defines the group as more than a collection of individuals, as they with their peers form a social system in which they experience interdependence, interaction and striving for a common goal[8]. Furthermore, she also argues that very few studies are available on group cohesion or on group norms. The study results support the claim that the student is greatly affected by his or her group, i.e. the level of autonomy. This means that if the group cohesion is good and the group norms are accepted, it raises the motivation of an individual member of the group.

Laperrousaz et al. have studied the perception of individual activities in a group activity through Qualitative Information about the Group Dynamics [9]; they have concentrated on the perceiving end, i.e. the tutor of the course. They have built tools that animate the level of collective activity, but their target group is different as students are distance learners and the staff support is different. However, such a visualization could be helpful if the amount of students rise above twenty, as the tutor will have a difficult time memorizing how each and everyone has acted in the event.

Mohan et al. have studied intra-group dynamics; they claim that it can be seen a major factor that influences team performance [10]. They suggest that improvements can be made by seeing it as a context-related factor. They also claim that the best results can be obtained through team cooperation. In the study, they have utilized the following parameters: i) acceptance of personal responsibility, ii) personal interests vs. group interests, iii) task orientation vs. social awareness, iv) leadership vs. follower, v) coherence and consistency of conceptual vision vs. cognitive diversity and valency, vi) need for flexibility vs. need for structure, vii) team conflict vs. conflict management.

There have been many research works on the benefits of collaborative and group work. In this paper, we were able to use only part of those while selecting criteria for the code camp.

3. ANALYSIS OF GROUP WORK EMPHASIS IN CODE CAMP PROCESS

The code camp process is evaluated on the basis of an example course. A .NET-programming focused course was organized as a code camp in January 2007. It was a week-long event in which the first day contained mostly introductory lectures. The introduction was as light-weight as possible in order to introduce the new software, software tools, platform, and the code camp as an event and also some get-together events such as a sauna evening. The second day started with demo sessions, and the actual programming started. The following days followed the same pattern. Not everything was shown immediately, as previous experience had taught us to proceed gradually even though we only had one week. On the last day, the students had to give a presentation as well a demonstration of their code. A rough outline of the schedule is presented in Table 2.

Table 2. Schedule of the code camp

Day	8-12	12-16	16-	Outcome
Mon	MS Tour	MS Tour	Sauna	
Tue	Intro	Demos	Planning	Plan
Wed	Demos	Program-ming	Program-ming	1st draft
Thu	Demos	Program-ming	Program-ming	2nd draft
Fri	Finalize the work	Presen-tation		Final

The learning outcomes of the code camp were related to NET-programming skills. However, the evaluation criteria also included some code camp specific aspects, such as the code camp spirit, i.e. helping others. The results of the Code camp were analyzed through student presentations and documentation as well as through a comprehensive course questionnaire. The course questionnaire consisted of 43 questions in 7 different categories; i) background information, ii) code camp as a teaching method (*situation*), iii) learning outcomes, iv) group dynamics (*process*), v) realization of the code camp, vi) .NET platform (*context*) and vii) future code camp courses. Out of 57 students, 40 returned the questionnaire. In particular, students were asked to evaluate how well different group dynamics related aspects were present in the code camp. These group dynamics were group cooperation (as a guideline, group cooperation was promoted so that groups would help each other), group interaction (as a guideline, group interaction

was promoted through being in same situation / environment and also by providing different tools such as wikis), role switch (as a guideline, role switch was promoted so that each and every person would learn the same learning outcomes), team spirit (as a guideline, the whole code camp method has been built on this attribute) and workload calculation (used as feedback for planning future code camp courses / tasks / etc.). Table 3 summarizes the results of the group dynamics category.
A five-level scale (from None to Excellent) was used for evaluation.

Table 3. Group Work questionnaire results (in %)

	None	Weak	Moderate	Good	Excel-lent
Group coop-eration	0	10	10	62	18
Group inter-action	0	15	41	44	0
Role switch	8	16	40	37	0
Team spirit	0	5	24	42	29
Work-load calcu-lation	5	18	33	36	8

The results presented indicate how the students felt about different group work aspects of the course. The results show that the cooperation within the group was rather good. This result indicates that the students were highly motivated for their group work. Group interaction was somewhere between good and moderate. Groups were actively advised to follow what other groups are doing. In addition to live discussions, a Wiki area was created for cooperation. All groups used the Wiki area to document their work as well as to share good web links and to ask and answer questions. Role switching (because every one needs to learn to produce code) was not as good. The current code camp processes do not emphasize project work such as role switching, and this can be seen in the results. The supervisor observation was that all students participated in all tasks, and thus the learning outcomes are the same for all of the students in the group. Students had evaluated the code camp / team spirit as good. This indicates high motivation and a will to help others in this type of a collaborative learning effort. Workload distribution measured the amount of work in the course and was also seen to be between moderate and good.

The results were also analyzed against the criteria given by Åberg [3]. The *group spirit* (code camp spirit), according to the student feedback, was mostly good or excellent. Actually, after every code camp (four so far), the students have been satisfied with the course. The group spirit is raised by the staff, which has shown solidarity. This is due to participation in the sleepless nights in a computer room. The students feel that the staff is 'in the same boat' and care about their learning. *Group norms* have been neglected or have not been consciously addressed in our code camps. Each student and each group has individual accountability, and if the students feel that e.g. 24-hour coding or collaboration is very strange and would rather operate differently, we have allowed that to take place. However, the commitment of the group to the joint task(s) is one of the criteria for the course, and that joint time also needs to be used to support others. *Group synergy* has not been optimized. The groups have not been built as heterogeneous as possible. This is another area of development that needs to be addressed in one way or another. As we use pre-tests to determine the skill level before the course, they could also be used to form groups. Groups where everyone knows something about everything are likely to gain from collaboration. If skills in the group are distinctly delimited, the persons might concentrate only on their own specialties and not learn from other members. However, this kind of development can be corrected by procedures where group members have to switch roles and e.g. evaluate each other's programming style. Groups may also be reassigned, in which case students have to be able to read completely new source code. This emphasizes good coding and commenting skills. This kind of an approach is presented e.g. in [11].

It should be noted that the questionnaire did not follow any clear preplanned or well-known methodology. Thus, even though the results appear to be mostly good, one could argue that the real relevance/utility falls short e.g. without clear definitions for what is good group cooperation. The literature provides partially appropriate definitions, but no quality check for making sure that the students really understood the terminology in the same manner was carried out. This aspect clearly needs to be improved in order to draw solid arguments. Thus, these results should be viewed in a discussional light, as e.g. their reliability cannot be methodologically justified.

As the pre-test was carried out only to determine the students' entry skills and not to form heterogeneous groups, it is very difficult to analyze the group dynamics in heterogeneous settings. This would also be very challenging, as mostly the courses are arranged once a year, and with different programming languages, software platforms, students in different phases in their studies, i.e. some are Bachelor's degree students and some Master's or even doctoral students. However, the skill level of the students rose, as Figure 3 in comparison to Figure 2 indicates. It must be noted that the results show an overall average rise, but do not show one on an individual level. Thus, it might be possible that the skill level for some students rose significantly, whereas for some students it could even have decreased. All values presented are based

on student perceptions. Figure 2 presents the students' perception of their skills at the beginning of the course, and Figure 3 at the end of the course.

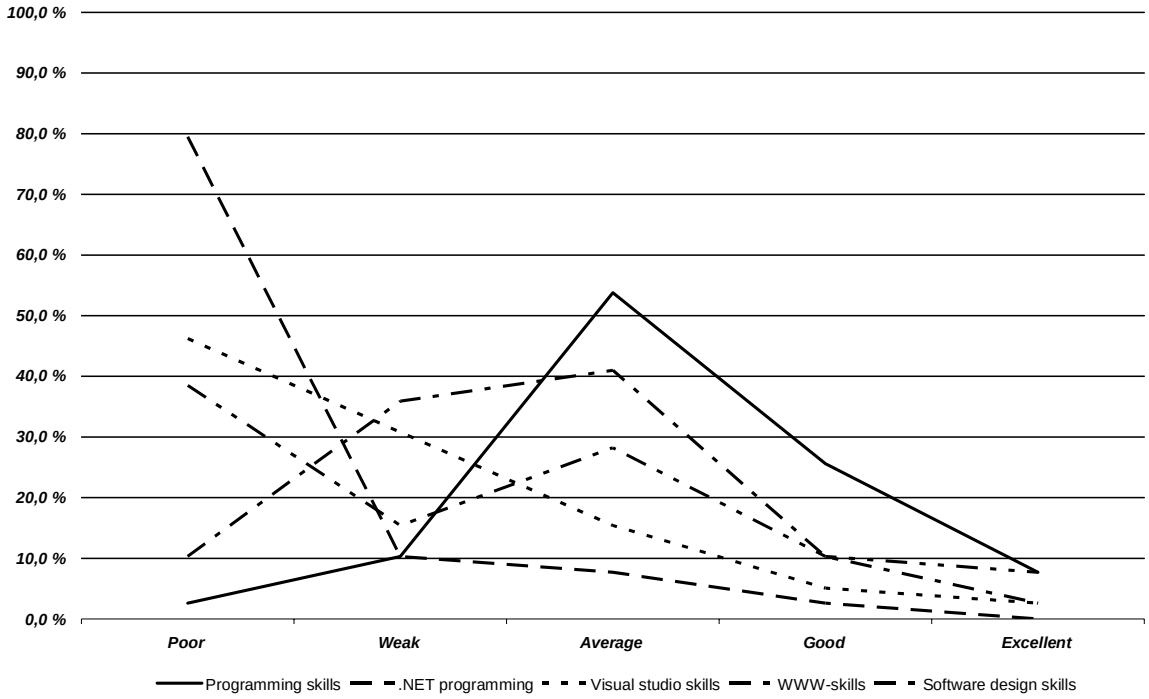

Figure 2. Skill development chart: skills before the course

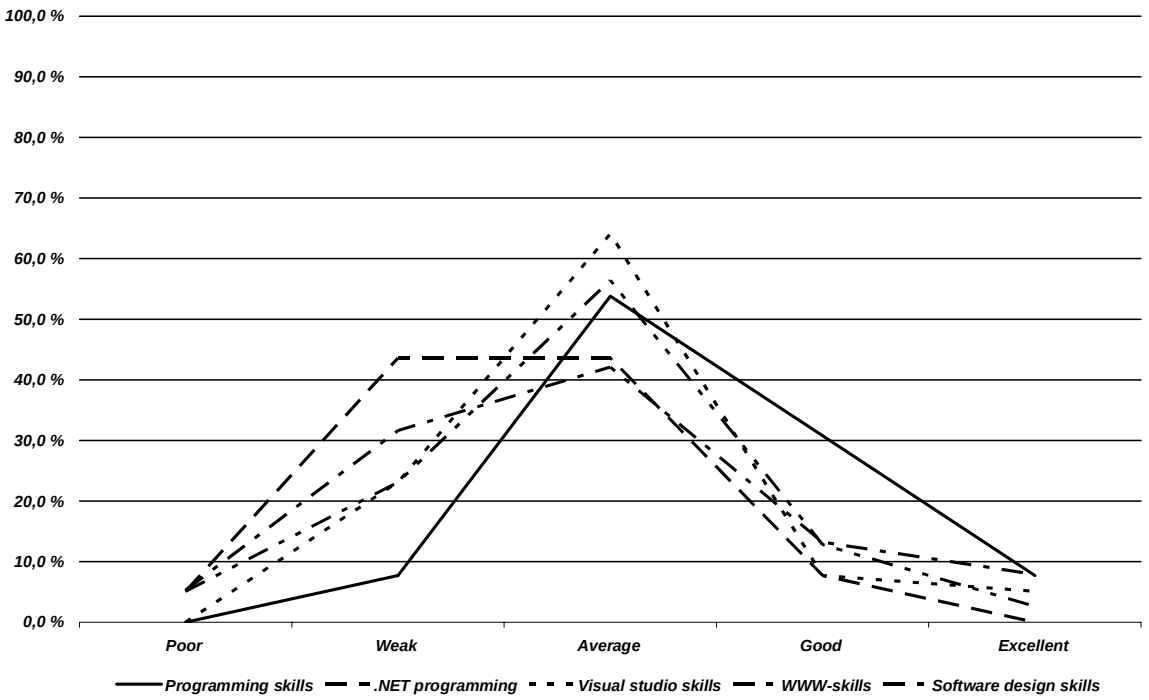

Figure 3. Skill development chart: skills after the course

Parameters presented by Mohan et al. could also be utilized to analyze the groups. Acceptance of personal responsibility is seen as a group issue, as there are currently no pre-set sanctions for students who drop out. The purpose of the get-together events is to get everyone on board and to feel comfortable. This is becoming more challenging, as more and more international students enroll for the courses. Cultural differences, e.g. the culture of doing things, need to be addressed. Personal interests vs. group interests are not addressed as well at the moment as we would like them to be. We do ask about this aspect in the course feedback, but that could be part of the pre-test inquiry, as well. However, as the students have no sanctions if they do not attend the course, the pre-test inquiry loses its value, as the final set of students might be different. This aspect surely requires its own more attention in future code camps. Task orientation vs. social acceptance is more guided than other parameters. The task is clearly addressed and the learning is more task or objective-oriented, and social acceptance is left for the groups. There might be a need for this, as more students with different backgrounds take part in the course. The leadership vs. follower parameter is not addressed, as we have preferred to have the groups act naturally. We have not asked for coherence in the group, nor that the groups provide data at a conceptual level. The main emphasis is on the operation of the code. The need for flexibility and control is a somewhat grey area; we have wanted to give all of the flexibility possible, and the only restrictions are deadlines. Thus we do not want to restrict creativity with too many rules. Team conflicts and conflict management are left to the teams. Most often the teams are rather busy and have no time for confrontations. In addition, the members cannot really afford them because helping others is one of the evaluation criteria.

Group activity is guided by the process. On a daily scale this is valid, but more accurate guidance for students might be helpful. It would support the group formation, reduce the storm in the group, the norming phase of the group, and the performance time, and finally, support the adjourning phase. The evaluation aims to promote the camp spirit and mutual assistance. However, a good grade can also be obtained by producing high quality code without inter-group interaction. Group activities need to be identified and analyzed in order to support them. Although the cooperative goal is clear, it is considered difficult. The feedback indicates that collaboration receives praise, but is yet very little utilized. That might be a cultural issue.

The aspects of learning together as seen by the students in the code camp are presented in Table 4. Students were asked how much they felt the following aspects affect the learning outcomes of the course: the intensive nature of the course, the course structure, the topic, the lecturer's expertise and the interactive nature of the course all had a positive effect on the learning outcomes. Only one's own expertise was not considered so important. It seems that the collaborative learning style in which the students stay together and interact does affect the learning outcome.

The code camps have been viewed as great fun or great experiences, but related learning is not yet scientifically proved, i.e. compared with groups using other study methods for the same material. The results indicate that students' skill levels have risen, but whether or not it is merely an impression or a lasting trend remains to be seen. After the sample course examined here, we have added different pre-tests and after-math tests to find out if the learning results are lasting.

Table 4. Aspects affecting learning (in %)

	None	Weak	Mode-rate	Good	Excel-lent
Time in classroom	0	0	18	56	26
Course structure	0	0	32	66	3
Topic	0	5	33	51	10
Own Expertise	0	37	34	24	3
Skills of the lecturers	3	8	29	61	3
Inter-activity	0	2	26	53	16

4. CONCLUSION

We have presented a code camp process that aims to utilize group dynamics and group processes. The work on the process itself is in its early stages, even though student feedback indicates that we have had several successful code camps in the past. The results of our sample course reveal that students experience the approach positively, but how much students actually learn during the camp is not yet scientifically verified. It seems that the code camp approach is a promising one, and we believe that by developing it further in terms of the understanding of group dynamics could benefit the actual learning even more.

REFERENCES

[1] Preston, D., Using collaborative learning research to enhance pair programming pedagogy, ACM SIGITE Newsletter, 3 (1), 2006, pp. 16-21.

[2] Porras, J., Heikkinen, K. and Ikonen, J., Code Camp: A setting for collaborative learning of programming, ACTA PRESS, Advanced Technology for Learning, Vol. 4, No. 1, 2007.

[3] Åberg, E. (in Finnish, leadership skills lecture series), Esimiesvietinnän luentosarja, Helsingin Yliopisto, Valitotieteellinen tiedekunta ja viestinnän laitos, 2002.

[4] Lindblom-Ylänne, S. and Nevgi, A., (in Finnish) Yliopisto- ja korkeakouluopettajan käsikirja, 2003. [in English, The University Teacher Handbook}

[5] Kovanen, K. (in Finnish, social psychology), Ryhmäprosessin merkitys oppimisessa, Pro Gradu tutkielma, Kasvastustiede, Tampereen Yliopisto, 2005, {in English, The meaningfulness of the Group processes in learning}.

[6] Joseph, A. and Payne, M., Group dynamics and collaborative group performance Technical Symposium on Computer Science Education, Proceedings of the 34th SIGCSE technical symposium on Computer science education, SESSION: Collaborative learning, pp. 368 – 371, 2003, ISBN:1-58113-648-X.

[7] Dooner, A-M, Mandzuk, D., and Clifton, R., Stages of collaboration and the realities of professional learning communities, Teaching and Teacher Education, In Press, Corrected Proof, Available online 26 October 2007.

[8] Ya-Hui Chang, L., The influences of group processes on learner's autonomous beliefs and behaviours, System 35 (2007), pp. 322-337, Publisher, Elsevier Publishing.

[9] Laperrousaz, C., Leroux, P. and Teutsch, P., Perception of Individual Activities in a Group Activity through Qualitative Information about the Group Dynamics, Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 732 – 738, 2005 ISBN:0-7695-2415-X Published by IEEE Computer Society.

[10] Mohan, A., Lemenager, E. and McCracken, M, Targeting Areas of Improvement in Intra-Group Dynamics using a Participative Approach - a Case Study, IEEE Engineering Management Conference, Sept. 2006 pp. 110 – 115.

[11] DeClue T. Pair Programming and pair trading; effects on learning and motivation in a CS2 course, Journal of Computing sciences in Colleges, Vol 18, Iss. 5, May 2003, pp. 49-56.