

Improving Student Group Work with Collaboration Patterns: A Case Study

Christian Köppe*, Marko van Eekelen[†], Stijn Hoppenbrouwers[‡],

*HAN University of Applied Sciences, Arnhem/Nijmegen, Netherlands

Email: christian.koppe@han.nl

[†]Institute for Computing and Information Sciences (iCIS), Radboud University Nijmegen and

Department of Computer Science, Open University of the Netherlands

Email: marko@cs.ru.nl

[‡]HAN University of Applied Sciences, Arnhem/Nijmegen, Netherlands and

Institute for Computing and Information Sciences (iCIS), Radboud University Nijmegen, Netherlands

Email: stijn.hoppenbrouwers@han.nl

Abstract—Group work skills are essential for Computer Scientists and especially Software Engineers. Group work is included in most CS curricula in order to support students in acquiring these skills. During group work, problems can occur related to a variety of factors, such as unstable group constellations or (missing) instructor support. Students need to find strategies for solving or preventing such problems. Student collaboration patterns offer a way of supporting students by providing problem-solving strategies that other students have already applied successfully.

In this work we describe how student collaboration patterns were applied in an interdisciplinary software engineering project, and show that their application was generally experienced as helpful by the students.

PLEASE NOTE: this is the authors' version of the work, to be published by IEEE in the Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering.

I. INTRODUCTION

Group work forms an essential part in computer science and software engineering education. The ACM/IEEE CS Curriculum Guide 2013 explicitly includes teamwork, communication, time management, and problem-solving skills as part of the 'software engineering and social issues' and 'professional practice knowledge' areas [1].

These skills are often included in curricula in student group projects, ranging from smaller projects to larger ones (such as a capstone project). Reports on what makes projects successful (in terms of high student satisfaction) include factors such as significant instructor guidance [2]. However, high instructor guidance (on how to work as a team) requires time and also may lead to students who expect the instructor to solve their occurring problems instead of addressing them actively on their own [3]. Having other ways of (partially) reaching the same goal might be helpful.

We believe that if we as instructors are able to support students in acquiring the required skills while having to invest less time in doing so, we may increase the satisfaction of students—and consequently, we hope, the quality of their group work—and also create the opportunity for instructors

to spend their valuable time on support of not group work-related issues .

Our main idea is that the usage of *patterns* for communicating well-known collaboration practices will have a positive impact on the acceptance and application of these practices. This especially because the presented collaboration patterns are based on the experience of, and described by, other students after completing multiple group projects and they are described in more detail—containing tacit and relevant knowledge—than most other already described practices.

In this paper we propose how student collaboration patterns can be introduced to support group work in education, and evaluate their application for a larger software engineering student project. The patterns themselves were described in earlier work and cover a range of good practices in student project work.

The contribution of this paper is to show educators and student project designers how they could improve student projects with respect to group work issues by including student collaboration patterns. By presenting an evaluation of the collaboration pattern application in a software engineering project, we offer information on which patterns were experienced as most valuable by the students and also which aspects they address.

II. RELATED WORK

Marques and Ochoa make use of ThinkLets [4] for transferring knowledge on how to solve team work problems. They describe ThinkLets “as a kind of process pattern to address collaboration recurrent problems”. The proposed ThinkLets address problems—internal and client-related—in the areas of communication, coordination, and motivation. They consist of a name, a short description of the recurring problem, possible corrective actions, and useful practices. These ThinkLets were given to students by student coaches—as part of supervising their team work—when problems occurred, and their application shows promising results. However, these ThinkLets are provided by student coaches *after* problems had been identified, so the students applied them after they ran into

problems. This is different from our approach, as we want to create an initial awareness of possible problems that can occur in projects and offer more generic solutions on how to solve them. However, a few of the described ThinkLets overlap with collaboration patterns as used in this work and the main idea—supporting students in learning teamwork skills through providing good practices—is similar.

Payne et al. researched which changes in group projects would make them more successful according to students' beliefs [5]. The findings were categorized in faculty centered and student centered themes, where the latter ones include communication as a value to improve group work, leadership and teamwork, and goal development. Among the more specific tips for other students were “Assign more specific tasks”, “Need to have defined task assignments for members”, “Clarity of goals”, or “Spread around responsibility”. Most of these can also (indirectly) be found in the presented collaboration patterns.

Hansen gives a suggestions for improving the performance of student teams based on a literature review [6]. Among these suggestions are “Teaching Team Development and Teamwork Skills”, “Establishing Clear Goals”, and “Requiring Team Members to Have Specific and Assigned Roles”. These suggestions, too, can be partly realized or supported by a few of the presented patterns.

III. STUDENT COLLABORATION PATTERNS

Patterns as a way of describing design knowledge originate in architecture and were introduced by Alexander et al. [7]. In the beginning of the 1990's patterns found their way into software development in the form of software design patterns, mainly introduced by the release of the famous *Design Patterns* book by the Gang of Four [8].

Patterns have been used in CS education for quite a while, too. The work of the Pedagogical Patterns Project—a substantial collection of patterns addressing various areas of (CS) education—played hereby an important role [9]. There are also patterns focusing on the students' perspective on learning [10].

Student collaboration patterns describe common solutions to problems that regularly occur in student group work projects. The goal of these patterns is to help students overcome common problems that likely occur during their first projects. The patterns themselves are based on the experience of other students who already took part in at least one larger project and were mined in a series of workshops. The mining of the patterns included a second set of goals, namely the inclusion of professional practice and competencies like reflection, analysing, and communication. For more details on those goals and the design of the pattern mining workshops, see [11].

The results of the pattern mining workshops were these eleven full patterns (published in [12], [13]): The complete list of patterns contained:

- 1) CLEAR UP QUESTIONS - ensure that there are no open questions when starting with the project.

- 2) SHARE EXPECTATIONS - Know what your teammates expect to get out of the project and what effort they will put into it.
- 3) GIVE A FIRST WARNING - React on malfunctioning group members in a constructive but clear way.
- 4) FILL KNOWLEDGE GAPS - If team members are missing knowledge necessary for fulfilling their tasks, then support them with acquiring this knowledge.
- 5) CENTRALIZE WORK PRODUCT MANAGEMENT - Ensure that all team members always have access to the latest project artefact versions.
- 6) MANAGE THE PROJECT - Define appropriate roles, tasks, and responsibilities for managing your project.
- 7) MEDIATE THE DISPUTE - Don't let disputes become real problems, but mediate them as soon and as constructive as possible.
- 8) KEEP MOTIVATED - If motivation to work on the project goes down, change tasks and work actively on being motivated again for the rest of the project.
- 9) START IMMEDIATELY - Don't lose valuable time by postponing to start working on the project.
- 10) REGULARLY CHECK REQUIREMENTS FULFILLMENT - Check regularly if the (intermediate) artefacts meet the requirements instead of waiting until the end, as the artefacts likely cannot be corrected by then.
- 11) SPREAD TASKS APPROPRIATELY - Ensure that all team members have an appropriate amount of tasks, according to their skills and responsibilities.

IV. RESEARCH QUESTION AND METHOD

The main research question of this work is:

Do collaboration patterns help students when working in project groups?

To answer this research question, we defined the following sub-questions:

- 1) SQ1: Were collaboration patterns generally perceived as valuable by the students?
- 2) SQ2: Were the patterns perceived as comprehensible and easy to apply?
- 3) SQ3: Which individual patterns were applied and how helpful was the application in the students' opinion?
- 4) SQ4: Which patterns were experienced as most valuable?

To answer the research questions, we conducted a survey study. We identified indicators that provide information necessary for answering the research sub-questions. The relationship between survey questions and research sub-question is given in the following survey description. Please note that we omitted a few questions that are not relevant for this work.

The first two questions were statements the students had to judge on a 5-point Likert scale (from 1=strong disagree to 5=strong agree). These statements were:

- Q1: The collaboration patterns were helpful in our project. (SQ1-indicator)

- Q2: The patterns were comprehensible and easy to apply. (SQ2-indicator)

Question 3 asked the students to declare per pattern (Q3a) if they had applied it, and (Q3b) when applied, how helpful they experienced it to be (SQ3-indicator). All patterns got an individual number in the given list as reference.

Question 4 was: Why were the 'not applied' patterns not applied? (Q4), it was intended to get more insight into the general pattern application.

Question 5 asked them to state all patterns (using the pattern numbers given earlier) which they would also apply in the next project when faced with a similar problem. If a student states that s/he would apply a certain pattern again, then it can be concluded that s/he sees value in applying the pattern. Q5 and Q3 combined formed the indicator for SQ4.

V. EDUCATIONAL CONTEXT

The student project was an interdisciplinary software engineering project at Radboud University Nijmegen, comprising third year undergraduate students of the Computer Science, Information Science, and Artificial Intelligence curricula. There were nine groups of six students each, a total of 54 students. Each group had a different software development project for which the architecture and the requirements were defined by other students in another course. The resulting software application was to be delivered to a real-life customer, i.e. a company or an institution that really needed the results of the project. Collaboration, i.e. contributing to the group activities and the project results in a synergetic manner, was an explicit educational goal of the project.

In the beginning of the project the eleven collaboration patterns (presented in Section 2.2) were introduced to the students and handouts were provided with the complete descriptions of the patterns. The students were reminded that collaboration was an educational goal and that the subject would also be part of the final written exam of the course. When appropriate, the teacher referred to the collaboration patterns in class.

All projects produced good results meeting all the 'Must Have' requirements. The marks on the collaboration question in the exam were in line with marks for other questions and marks in other years.

VI. RESULTS

In total 43 students (out of 54 project participants) responded to the survey, which was taken two months after the project started and the collaboration patterns were introduced.

The mean answer to question Q1 ("patterns helpful in project") is 2.86. It is obvious that the students slightly disagree that the patterns were helpful in their project. However, the numbers of patterns the students would apply again and the numbers in Table I suggest that at least some of the patterns were experienced as helpful. Our interpretation is that these patterns of course only cover a small aspect of the project, namely the group work issues. They do not help in technology-related issues. The students might have answered this question with the whole project in mind and not only the group work

TABLE I
APPLICATIONS PER PATTERN (Q3A, N=43); PATTERN APPLICATION HELPED (Q3B): 1=NOT, 2=A BIT, 3=A LOT (ONLY APPLIED PATTERNS TAKEN INTO ACCOUNT); WOULD APPLY PATTERN/S AGAIN (Q5, (N=43))

Pattern	Q3A #	Q3A %	Q3B mean	Q5 #	Q5 %
Clear Up Questions	32	74.4	2.28	17	39.5
Share Expectations	27	62.8	2.15	17	39.5
Give A First Warning	10	23.3	1.80	2	4.7
Fill Knowledge Gaps	29	67.4	2.38	16	37.2
Centralize Work Product Mngmt.	16	37.2	2.19	4	9.3
Manage The Project	23	53.5	2.08	7	16.3
Mediate The Dispute	12	27.9	1.83	7	16.3
Keep Motivated	24	55.8	1.83	11	25.6
Start Immediately	27	62.8	2.33	22	51.2
Regularly Check Reqs. Fulfillment	22	51.2	2.10	7	16.3
Spread Tasks Appropriately	33	76.7	2.45	18	41.9

aspects, which would have led them to a slightly negative reaction.

The answers to Q2 ("comprehensible/easy to apply.") show a mean of 3.6, which suggests that the way the patterns were presented was easy to understand for the students.

On average, 5.93 patterns were applied per student (N=43, median=7.0, mean=5.93). 8 students did not apply any pattern (most of them stated that they forgot to do it), if we exclude these the mean is 7.29 (N=35).

The number of applications per pattern might not be directly comparable, as some of the problems addressed by the patterns are nearly always present (e.g. START IMMEDIATELY or SHARE EXPECTATIONS) while others are only applicable in specific situations (e.g. GIVE A FIRST WARNING when slackers are present). However, all patterns were reported multiple times as being applied.

Table I shows per pattern the number of mentions which of them would be applied again in a next project (Q5). We only counted the patterns that were explicitly mentioned by number. However, 8 students mentioned that they likely would apply most or all of them if appropriate. Adding these to the numbers would increase both # and % by 8 and 18.6 respectively.

Table I indicates that a few patterns were experienced as most helpful and that the same patterns also were mentioned most often in the question which patterns the students would apply again. These patterns (highlighted in Table I) are CLEAR UP QUESTIONS, SHARE EXPECTATIONS, FILL KNOWLEDGE GAPS, START IMMEDIATELY, and SPREAD TASKS APPROPRIATELY.

VII. EVALUATION AND DISCUSSION

Based on the results of the survey, we can answer the research subquestions as follows.

SQ1: Were collaboration patterns generally perceived as valuable by the students?: The students answered this question (Q1) directly with a slight disagree. Contrary to this direct answer, 91% of the students stated that they will at least apply a few of the patterns in their following project again (Q5). 8 students even stated that they will apply all of the patterns when applicable. We interpret this as indication that

the students implicitly did perceive the collaboration patterns as valuable, but do not express this explicitly.

SQ2: Were the patterns perceived as comprehensible and easy to apply?: Yes, the answers indicate that most students had no problems understanding and applying the patterns. This is supported by the high application number of some of the patterns (up to about 75%). There were also no indications in the answers on Q4 (“Why were the not applied patterns not applied?”) that understandability or applicability were issues for the students.

SQ3: Which individual patterns were applied and how helpful was this application in the students’ perception?: All patterns were applied by the students. Eight of the eleven patterns were applied by more than 50% of the survey respondents. Some of the applied patterns were experienced as more helpful than others, these are highlighted in Table I. Because we wanted to keep the survey relatively short, we did not gather information on *why* patterns that were experienced as more or less helpful. This will be part of future work.

SQ4: Which patterns were experienced as most valuable?: The answers on Q3b (“How helpful were the applied patterns?”) already indicated which of the applied patterns were experienced as valuable. Table I shows numbers on which patterns the students would apply again, which is an indirect indicator that these patterns are seen as valuable. The patterns mentioned most often are the same patterns as mentioned in the previous question, so based on the survey responses these are also the patterns which are seen by the students as most valuable.

Based on the subquestion answers, the main research question “Do collaboration patterns help students when working in project groups?” is answered as follows: According to the results, the students experienced the application of some of the presented collaboration patterns as useful. This experienced usefulness is less strong if the whole project is taken into account.

The patterns CLEAR UP QUESTIONS, SHARE EXPECTATIONS, FILL KNOWLEDGE GAPS, START IMMEDIATELY, and SPREAD TASKS APPROPRIATELY can be seen as the most important or valuable patterns in the collection. They share certain characteristics: all are important for—and applied during—the beginning of a project and they address issues which always occur in group work. The less valued patterns are patterns that are more relevant towards the end of the project or address issues which only occur occasionally (like disputes in the group or the presence of slackers).

The second indication that the patterns were experienced as useful is that 91% of the students indicated that they will at least apply a few of the patterns in their following project (8 students even stated that they will apply all of them when applicable).

VIII. CONCLUSIONS AND FUTURE WORK

The results of this case study show two results: some of the practices described in the patterns were experienced by the

students as valuable addition to their projects. This is indicated directly by the answers on question Q3B and indirectly by the answers on question Q5 (students will apply them again in future projects).

However, the presentation of these practices in the form of patterns was not experienced as either helpful or not helpful by the students (see question Q1), but still mostly comprehensible and easy to apply (question Q2). Our interpretation is that the information contained in the patterns and the way they are presented is more important than the fact that these are patterns. This assumption needs to be proven in future work, but could lead to a possible change in the way the patterns are presented to the students.

Another change that can be made for the presentation of the patterns is a prioritization. The most mentioned/valued patterns should get the main focus or become obligatory patterns, while the other patterns only have to be applied when necessary. The effects of such a prioritization need to be determined in future work. Furthermore, we plan to research how the students applied the patterns and if the patterns also work in domains other than software engineering.

REFERENCES

- [1] Joint Task Force on Computing Curricula ACM/IEEE, *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc, Jan. 2013.
- [2] B. A. Oakley, D. M. Hanna, Z. Kuzmyn, and R. M. Felder, “Best Practices Involving Teamwork in the Classroom: Results From a Survey of 6435 Engineering Student Respondents,” *IEEE Transactions on Education*, vol. 50, no. 3, pp. 266–272, Aug. 2007.
- [3] D. Damian and A. Borici, “Teamwork, coordination and customer relationship management skills: as important as technical skills in preparing our SE graduates,” in *Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences, EduRex’12*. IEEE Press, Jun. 2012, pp. 37–40.
- [4] M. Marques and S. F. Ochoa, “Improving teamwork in students software projects,” in *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, Apr. 2014, pp. 99–108.
- [5] B. K. Payne, E. Monk-Turner, D. Smith, and M. Sumter, “Improving Group Work: Voices of Students,” *Education*, vol. 126, no. 3, pp. 441–448, 2006.
- [6] R. S. Hansen, “Benefits and Problems With Student Teams: Suggestions for Improving Team Projects.” *Journal of Education for Business*, vol. 82, no. 1, pp. 11–19, 2006.
- [7] C. Alexander, S. Ishikawa, and M. Silverstein, *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, Aug. 1977.
- [8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: elements of reusable object-oriented software*. Boston, MA: Addison-Wesley, 1994.
- [9] Pedagogical Patterns Editorial Board, *Pedagogical Patterns: Advice for Educators*, J. Bergin, Ed. New York, NY, USA: Joseph Bergin Software Tools, 2012.
- [10] T. Iba and M. Sakamoto, “Learning Patterns III - A Pattern Language for Creative Learning,” in *Proceedings of the 18th Conference on Pattern Languages of Programs, PLoP’11*. Portland, OR, USA: ACM, 2011.
- [11] C. Köppe, “Using pattern mining for competency-focused education,” in *Proceedings of Second Computer Science Education Research Conference, CSERC ’12*. Wroclaw, Poland: ACM Press, 2012, pp. 23–26.
- [12] C. Köppe, “Learning Patterns for Group Assignments - Part 1,” in *Proceedings of the 19th Pattern Languages of Programs conference, PLoP’12*. Tucson, Arizona, USA: ACM, 2012.
- [13] T. de Cortie, G. Bosma, R. Broeckhuijsen, and C. Köppe, “Learning Patterns for Group Assignments - Part 2,” in *Proceedings of the 20th Pattern Languages of Programs conference, PLoP’13*, Monticello, Illinois, USA, 2013.