Exploring Software Engineering Catalogues with Poster Sessions

Barne Kleinen, Debora Weber-Wulff {kleinen|weberwu}@htw-berlin.de International Media and Computing HTW Berlin Germany

Abstract—In modern software engineering there are many aspects of software craftsmanship that consist of long catalogues of patterns or of dos and don'ts. For example, there are a number of standard books on important topics such as refactorings, aspects of Clean Code and design patterns that are useful to the practitioner, but would be extremely boring if they were to be taught by an instructor displaying slide after slide. Encouraging students to produce and present posters on these topics can be a useful pedagogical method for teaching these subjects. This short paper will discuss the use of poster sessions in university-level software engineering courses.

I. INTRODUCTION

Traditional software engineering instruction at university has often been restricted to a lecture/exercises/project format. After listening to a lecture on theoretical aspects of software construction, the students solve either small exercises, or work in a group on a larger software project. Among the concepts that need to be understood are those concerned with large catalogues of related concepts. There are standard reference books available for at least three of these concepts: Refactoring [1], Design Patterns [2], and Clean Code [3].

However, while these books give quite detailed information about a wide range of aspects sorted by various criteria, teaching these catalogues of patterns is a chore if one resorts to only presenting slide after slide of the various items. On the other hand, if only a few items are chosen from the catalogue to illustrate in detail, it is difficult to impart a wide sense of the possibilities these catalogues do offer.

The authors have successfully used a number of variations on setting up poster sessions with students about these topics over the past few years and will be describing this teaching and learning method in detail in this paper.

II. DIDACTICAL BACKGROUND

The concept of microteaching was developed in the 1960s at the Stanford Teacher Education Program by Dwight W. Allen and Arthur W. Eve [4]. This was a method of teacher training that focused on taking turns teaching short snippets of content in a fixed time to fellow students. Over the years this has been found to be a very effective teaching method. From 2006 this concept was expanded by Peter Hug *et al.* to include the perspective of the learners, termed microlearning [5], and expanding it beyond teacher training. The focus of microlearn-

ing, which Hug defines to be, among other aspects, a relatively short time spent addressing very small content units ("knowledge nuggets") as part of a curriculum in various mediated learning situations [5, p. 9].

Hug's focus was on developing microlearning concepts for e-learning situations, but it can just as well be applied to other innovative teaching methods that address the learning of "knowledge nuggets" such as those found in the catalogues mentioned above. In particular, poster construction and presentation can be used as such a microteaching and -learning method, although it does have a whiff of use only in K-12 instruction [6] and are not in widespread use at university level. Poster sessions are, however, offered at many computer science conferences as a means of informing fellow researchers about current research that is perhaps not quite ready for peerreviewed publication. Thus it is an important communication method that should be practiced at undergraduate level.

The authors of this paper are professors at the University of Applied Sciences HTW Berlin in the International Media and Computing programs. Both a bachelor's degree and a master's degree are offered. A three-semester series of courses is taught in the bachelor's program, consisting of "Introduction to Programming", "Algorithms and Data Structures", and "Software Engineering". All three classes consist of four 45-minute hours of seminaristic instruction/lecture a week, usually comprised of two sessions of 90 minutes each, along with a two-hour session in a computer laboratory. There are as a rule approximately 40 students in a cohort, which are split into two groups of 20 for the laboratory sessions.

Among many other topics, the students need to have knowledge of refactoring, the basic design patterns, and the tenets of Clean Code by the end of this series and should be able to apply them appropriately when coding. All three topics are based on a primary or important resource book with which the students should be acquainted: Refactoring [1], Design Patterns [2], and Clean Code [3]. These particular books may not be the original source of most of the concepts presented in them, but they provide an important overview and catalogization of these topics that programmers need to refer to regularly and apply in their work.

These topics are also similar in the sense that it is necessary to see and apply them in practice to really grasp and understand them. While there are examples for refactorings, patterns and clean code issues in the lab exercises and lectures beginning with the first course, only a fraction of them can readily be applied in the exercises due to the small scope of the exercises. The poster sessions are not the only contact students have with these topics, but supplement the general overview given in lectures. The poster sessions foster discussion among the students about the application and applicability of, e.g., a pattern or refactoring similar to the design discussions done in a real software project, in which the different possibilities of applying the guidelines are carefully weighed against each other.

As the poster sessions are supplemental to an overview of the topic given in a lecture, it is not necessary to cover all catalogue items within a poster session. Students do, however, obtain an overview of the whole catalogue and dive deeper on a couple of topics. A wide selection of catalogue items is offered and it is left up to student's curiosity which ones are covered. It is hoped that this will spark their interest in learning more, as they are now aware that there is a larger catalogue they need to study more if they want to become professional software developers.

III. CONDUCTING THE POSTER SESSIONS

This section will discuss the various types of poster sessions in software engineering instruction used by the authors for third-semester students in computing.

A. Preparation Phase

Before the actual poster sessions start, it is necessary to give the students time for preparation. They are to work in groups of two to four persons, depending on the size of the class and the number of topics available.

The relevant chapters from the books need to be copied so that they can be handed out to the students. A list of the potential topics is prepared so that as the groups are formed, they can choose on which topic they wish to work.

Materials need to be readily available for poster construction, especially large sheets, coloured markers, sticky notes in various colours, coloured ovals and rectangles from moderation materials, scissors, tape, and glue.

Depending on the topic, the authors have worked with two main strategies for the poster preparation phase: Either completely in class within one day, e.g., in a double session (2 x 90 minutes) or as homework for the next class session. If poster preparation and poster presentation are not done on the same day, we found it necessary to assign points affecting the overall grade for preparing and showing up with a poster in order to ensure sufficient attendance for the poster presentation day. When both preparation and presentation are done on the same day, students usually follow through and prepare and present a poster even without an effect on the grade.

In either case, the students can be given a short introduction to the topic and given 45-60 minutes of class time for preparation.

For the topic of Clean Code, it was found to be necessary to give the students ample preparation time and that poster quality and depth was better when prepared as homework. This can also help students who have difficulties reading an English text, because they can invest more time in understanding the material.

Fowler's book on refactoring [1] describes more than 90 refactorings and 22 code "smells", indications that code might be problematic or cause errors in the future. Students already know the basics of refactoring as well as some of the refactorings; thus, students are usually quick to understand one single refactoring. This makes the topic of refactoring quite suitable for a shorter preparation phase making it easy to accommodate it even in one 90-minute session. Additionally, the catalogue of refactorings is available online [7], making it unnecessary to prepare any printed hand-outs beforehand.

With the design patterns, we started out with poster sessions based on copies of the relevant pages from the original Gang of Four Book [3], similar to the Clean Code poster session. However, it seems that students find it easier to understand the patterns if they refer to the more modern descriptions and examples abundantly available online, e.g., [8–10], which also eases preparation for a poster session for this topic.

B. Topic Choice

For a class of 40 students, they can be paired off into around ten to fifteen groups of two to four persons. Depending on the presentation format planned, it may or may not be important to have a certain or usually limited number of groups.

Martin Fowler's Clean Code book [2] can be nicely split into at least ten topics. Some of the chapters are rather large, but can be split into two or three portions, so that each group has approximately ten pages to read and report on.

The following chapters on Clean Code are well suited to poster making: 2 (Names), 3 (Functions, 2 parts), 4 (Commenting), 5 (Formatting), 7 (Error Handling), 10 (Classes), 13 (Concurrency), 17 (Smells and Heuristics, 3 parts).

The list of refactorings is rather long and while they could be characterized by frequency of use, it was decided not to assign them priorities as it may actually be more useful to cover less frequently used ones, as the students already know some of the frequently used refactorings. Therefore, in general, a random assignment of the refactorings was used.

Depending on the size of the group, a number of items were randomly assigned to the group using a Ruby script, generally around seven. Students were then asked to choose two or three refactorings from the assigned set and were asked to create "mini-posters" on regular DIN A4 paper using felt-tip pens. This way, a wide variety of the refactorings was covered, partly chosen at random and partly by the curiosity of the students. In addition, by being able to choose two out of seven refactorings, students who struggled to understand the concepts had the possibility to choose easier refactorings, while others were able to tackle the more challenging ones in their list, or to cover more refactorings.

IV. PRESENTATION FORMAT

The posters can be presented in a variety of formats. For all of the formats it is, however, necessary to have materials available for attaching the posters to the walls. Some institutions frown on tape being used, as this can remove the paint from the walls, so it is important to know the rules of the university on hanging up posters. Having a number of rolls of wide, clear cello tape, removable mounting putty, and sufficient magnets on hand for the posters to be put up in parallel is quite important. If there is not enough wall space, a cork-board on wheels can be rolled it and set perpendicular to a wall so that both sides can be used. Students must be on time to get their posters up and be ready to go as the presentations begin. There are three major types of presentation that we have experimented with.

A. Speed Dating Poster Session

The idea for a speed dating poster session came from a Wikimedia barcamp at which twelve wiki-based projects were presented. Each project had either a poster or a computer with a number of tabs open, and one presenter. The rest of the persons in the room distributed themselves around all of the projects, and the presentations began. The presenter had 3.5 minutes for explaining the wiki project, then one minute for questions and half a minute for the visitors to move on to the next poster. A moderator rang a bell in order to keep things moving. After an exhausting hour, all visitors had seen 12 projects and the presenter had presented 12 times, but unfortunately the presenter was not able to see or discuss any of the other posters.

This shortcoming can easily be overcome if there are at least two persons responsible for one poster. The posters are put up on the walls around the classroom in a ring. Each stands in front of their poster at the beginning of the session. The team splits up into two parts (1-1, 1-2, 2-2), depending on the team size, called the *travellers* and the *presenters*. The presenters stay with the group poster and will be presenting the poster in the first round. The travellers will be visiting the other posters, in either clockwise or counter-clockwise direction as determined by the instructor. Before the first presentation begins, all traveller groups move to the next poster in the given direction.

The presenters stay at the team poster and now explain the poster to each group as they come by for a few minutes. Depending on the number of posters and the time available for the presentation, this may be between three and five minutes. The organizer keeps time with a stopwatch and uses a bell or gong to signal that it is time to move on. 30 seconds should be scheduled for saying "thank you and good-bye" and moving on to the next station.

After all the other posters have been visited, the travellers are back at their home poster, and now the roles are switched for a second round. This session, too, begins with the travellers moving to the next station.

At the end of the second session, each student will have explained their own poster many times, and heard the presentations from each of the other stations. Having the presentation be both aural and visual promotes a deeper understanding of the material, instead of just skimming through the book text.

B. Wall of Mini-Posters

This presentation format is better for topics such as refactoring that are characterized by many smaller units. Each poster is much smaller, both physically as well as content-wise, so even a five-minute presentation would be too long for them. There are also far too many topics for a more formal approach.

These posters should fit on regular DIN A3- or DIN A4size paper and can be put up around the room at eye level. With the small format, 30 to 40 refactorings presented as miniposters fit nicely on a long wall, providing a wide overview over around a third of the refactorings. There is no defined order or clocking to the viewing. As in a museum, the students are welcome to read and discuss the posters with their fellow students in any order and for as long as they want. The instructor reads all of the posters as well, praising good ones, discussing the topics with student groups, and perhaps asking questions about those that don't quite make sense. Of course, there are some students who use this time to surf the Internet or catch up on WhatsApp, but most of the students enjoy seeing what the others have done and participate actively in discussions. Through this exercise they obtain a sense of the breadth and depth of the catalogue and are now better prepared for selfdirected study of the topics.

A variant that we are planning for the future is to have every student choose five posters that they find most important and have them write a short summary of each as a hand-in work product to be graded.

C. Ad-Hoc Poster Sessions based on Online Research

Poster sessions can also be held ad-hoc based on online research. This might be necessary if the energy level in the classroom is low or the weather too hot and a more engaging format than a lecture is called for, although a lecture had been planned. There are several topics which work well with such spontaneous poster sessions, as the subject is clearly structured and enough high-quality documentation is available online, enabling the students to start researching their topic right away, given that they now generally have Internet access from their own devices directly in the classroom. We have conducted poster sessions that were researched online for sorting algorithms, the Java Collections API, design patterns, and general programming topics as an exam review.

These topics are clearly structured and well known to instructors, so spontaneous guidance can easily be offered. A grouping of the students and assigning them to topics can easily be done ad-hoc on a whiteboard or in a projected text file. By choosing the most relevant topics out of the list, the number of groups can easily be adjusted.

V. EXPERIENCE

There are a number of issues that have arisen from the work with posters that will briefly be discussed here.

- The grading for a poster session should be all, half or nothing, the posters themselves should not be assessed, as not every student is good at graphic design. If a student participates in making and presenting a poster, they should have full credit, half credit if they only make or only present. Those who refuse to take part are given zero points for this exercise. Generally, this could be up to 5% of the final grade for a course, or used as extra credit.
- Using poster-sessions for teaching are not in widespread use at university level. This may be due to the much larger class size in tertiary educational settings. In our experience, class sizes between 30-45 students work well with the approach presented, which correspond with class sizes at German universities of applied sciences (*Fachhochschulen*). However, this approach probably does not scale to larger classes unless these classes are divided up into groups. Group sizes working on one topic could be made larger, but that increases the probability of a few doing the work for the entire group It also does not scale to smaller groups, as there must be a minimum number of students to cover enough variety

to actually have a poster session covering a good sample of the topics within a catalogue.

- Posters that are assigned as homework more often produce quite creative and even interactive posters with moving parts (windows or sliders). Most posters are of a quality that explains the concept without further explanation necessary.
- Even without a formal structure such as "speed dating" the students engage in lively discussions in front of the posters.
- Students alternate between self-study, reading the posters of others, explaining their own posters and questioning other posters. Together with the physical movement this creates a multi-modal learning experience, which is engaging and accommodates different learning types.
- Even if the students do not read and understand all of the posters, they get a sense of the variety of the subject and practice in reading the original material.
- In exams students have been asked to write down three aspects of Clean Code with the reasoning behind the tenet. Students remember their own section very well and are usually able to explain two more in good detail. The only ones unable to answer this question at all are generally those who did not attend the poster session.
- The teams should be assigned at random. In our experience, newly, randomly created teams work on the assignments more thoroughly, compared to just letting them work with their favourite teammate or seat neighbour. Additionally, the necessity to move in the classroom to draw and build these teams constitutes a starting momentum for switching from the lecture or introductory part to active teamwork.
- It is better not to do more than one or two poster sessions per class semester, as the novelty wears off fast.
- A student recently reported that the company where she works wanted to quickly learn about Clean Code. They ended up doing speed-dating posters, and now the posters adorn the walls in the development department, as constant reminders about writing good code. For those

who program, posters prepared by a group is a good way to read and study a book in parallel.

VI. CONCLUSION

Poster sessions have been found to be a valuable method for teaching software engineering subjects that cover a catalogue of similar things. It combines various activities in the classroom, as reading, writing, drawing and manufacturing the posters (coloured pens, scissors, and glue are often used), discussion, presentation, and debate are combined with a lively movement in the classroom. Last but not least, the students are encouraged to explore the material themselves in-depth and obtain practice in doing so.

REFERENCES

- M. Fowler, Refactoring. Improving the Design of Existing Code. Boston: Addison Wesley Longman, 1999.
- [2] R. C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship (1 ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA. 2008
- [3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [4] D. W. Allen and A. W. Eve, "Microteaching," Theory Into Practice, Vol. 7, No. 5, pp. 181–185, Dec. 1968.
- [5] P. Hug, "Microlearning: A New Pedagogical Challenge (Introductory Note)," In P. Hug, M. Lindner, & P. A. Bruck (eds.), Proceedings of the Microlearning 2005:: Learning & Working in New Media Environments June 23-25, 2005, Innsbruck. Published by Innsbruck University Press 2006, pp. 7–11.
- [6] J. O. Osa and L. R. Musser, The Role of Posters in Teacher Education Programs. Education Libraries, 27(1), 16-21, 2004. Available at https://k12teacherstaffdevelopment.com/tlb/how-can-i-use-postermaking-to-facilitate-learning-in-the-classroom/
- [7] M. Fowler, Catalog of Refactorings, 2013. Available at https://refactoring.com/catalog/
- [8] Refactoring Guru. Available at https://refactoring.guru/designpatterns/catalog
- [9] Design Patterns in Java Tutorial on tutorialspoint. Available at https://www.tutorialspoint.com/design_pattern/index.htm
- [10] P. Hauer: Design Pattern. (in German). Studienprojekt 2009-2010, Available at https://www.philipphauer.de/study/se/design-pattern.php