Experience-Oriented Approaches for Teaching and Training Requirements Engineering: An Experience Report

Andrea Herrmann[1], Anne Hoffmann[2], Dieter Landes[3], and Rüdiger Weißbach[4]

[1] Herrmann & Ehrlich, Stuttgart, Germany
herrmann@herrmann-ehrlich.de
[2] Siemens AG, Erlangen, Germany
anne.hoffmann@siemens.com
[3] University of Applied Sciences, Coburg, Germany
dieter.landes@hs-coburg.de
[4] Hamburg University of Applied Sciences, Hamburg, Germany
ruediger.weissbach@haw-hamburg.de

Abstract. [Context & motivation] Experience-oriented learning is known to be more efficient than learning by listening. Small team projects can teach practical issues of applying methods and soft skills. **[Question/problem]** RE is a core qualification for diverse stakeholders, not only for software engineers. In trainings and academic education, people with different professional backgrounds and different experiences, representing different stages in the Dreyfus model of skill acquisition, come together. The teaching's setup should take this into account. **[Principal ideas/results]** This experience report presents examples of various approaches for teaching RE in academia and industry. We discuss findings from interdisciplinary projects and game-oriented approaches, differences of these learning settings and differences which are to be considered when designing didactic settings for different target groups. **[Contribution]** This article presents diverse course concepts and experiences, and shall inspire other instructors to seek for additional learning approaches by taking into account their participants' heterogeneous background.

Keywords: Dreyfus model, experience-based learning, experience-oriented learning, interdisciplinary, learning, requirements engineering, teaching, training

1 Motivation: Why Teaching and Training Requirements Engineering?

Relevance of Requirements Engineering: Requirements Engineering (RE) is known to be critical for the success of software projects [1] and the education of software practitioners [2]. While this is a good motivation for learning and teaching RE, the growth of RE-related conferences and the rising number of RE certifications show that RE training has been recognized and accepted as being important.

C. Salinesi and I. van de Weerd (Eds.): REFSQ 2014, LNCS 8396, pp. 254–267, 2014.

Yet, RE knowledge is not only relevant for the software development team, but also for customers and other stakeholders, especially in the context of agile development. For instance, we see an increasing importance of business departments in decisions on information systems. This is not only caused, but intensified in the context of Cloud Computing, where business departments are able to buy Software as a Service (SaaS) products [3]. This trend is reflected in the authors' "experience" in teaching and training RE not only for IT specialists, but for marketing and engineering staff.

Whilst RE is typically seen as a task in an early stage of a waterfall project, RE is part of agile development, too.

Teaching vs. Training: We distinguish between teaching and training:

"Teaching" means the education that is part of the university curriculum during the studies and that prepares students for diverse work environments and roles. Most of the students have no work experience. Therefore, teaching cannot presuppose specific previous experiences or problem-awareness.

In our understanding, "Training" is defined as commercial training given to professionals. It is typically customized to the "target group's environment", to the role (developer, project manager, etc.), and to the domain (e.g. automotive), and takes company- or domain-specific standards into account. Most participants in trainings have work experience and attend the course with specific questions in their minds which they seek answers for.

In this experience paper, we try to find some answers to the question: "Which approach is appropriate for teaching or training RE to participants on which Dreyfus level?" This paper is founded on the practical experiences and the reflections of the authors compiled in diverse RE teaching and training situations.

2 Related Work

2.1 Four Forms of Experience-Oriented Learning Methods

"According to Lethbridge's survey" [2], software professionals think that their education has been moderately relevant for their job (3.5 points on a scale of 0 to 5). From their point of view, it was more important to learn how to think than to learn specific methods.

It is known that learning by listening is not as efficient as learning by doing [4]. While sitting in a lecture and **listening to the lecturer**, the student learns facts. However, only by **applying a method**, they get experience and learn the soft facts which cannot be transferred easily by lecturing.

For instance, learning the elements of an UML model is only the first step in learning UML. When applying the method to a simple toy example, already many practical questions arise. However, applying UML modeling to more complex systems gives rise to further questions, like the ideal level of granularity or how to check for completeness. When students **run a project** from beginning to end, building on the requirements which they elicited and modeled themselves, they get

direct feedback about the quality of their requirements. **Working in a team** additionally allows them to train soft skills they will need in practice. In distributed teams the students learn to work in virtual teams [5]. The most realistic learning effect can be achieved when students work in a **real-life project** with real organizations. A real-life project, compared to a role game, introduces difficulties beyond applying the rules of the RE methods. As the course has a certain duration and speed, it is not always easy to define a real-life project with the suitable scope.

Role games or project simulations are conducted in a protected and controllable university environment. Consequently, for instructors these approaches are often an ideal compromise between applying methods to unrealistically simple examples and complex and barely controllable real-life projects.

We define the difference between "role game" and "project simulation" as follows: Role games are a collection of individual games like an interview or an example where different methods are applied. In project simulations, methods and roles are applied to the same project. The teacher controls the role game or simulation by defining clear rules for the roles´ interactions during each step of the game.

Examples of role games in RE teaching are:

- The agile hour [6] and the extreme hour [7] where within one hour the learners simulate an agile project with three iterations, with the objective to implement a result by drawing it.
- Simulated customer interviews and subsequent development tasks [8], [9], which can be combined with improvisational theatre [10].
- A business game where students create software companies and bid for a large scale development project [11].
- Performing a development project which leads to a user interface design [12] or to a prototype software [13].
- Simulating a software project in the form of a card game [14], [15].

Additional roles games within different types of engineering and natural science courses are discussed by Fadali et al. [16].

Compared to real-life projects, games are shorter interactions within a canonized set of rules and under controlled circumstances. Dawson [17] even recommends to play tricks on the students in order to better prepare them for work life. Such tricks can be: to present an uncertain and naive customer, to change requirements and priorities, and to have conflicting requirements and pressures. In real-life projects, the longer duration increases the possibilities of interventions and disruptions.

Another way of learning is to analyze and discuss case descriptions, either real or invented cases. Cases are derived from realistically large projects in a way to highlight a specific aspect that is to be learned. Thus, students can learn from real projects, without the risks involved in executing them themselves.

Improvisation theatre was invented by Keith Johnstone [18] to support students in acting and drama improving their acting abilities. Johnstone invented so-called games each training certain aspects of communication and self-awareness. The REIM approach utilizes storytelling to map these games to typical Requirements Engineering situations, see [19, 20] for details.

2.2 The Dreyfus Model

The Dreyfus model, introduced by Stuart and Hubert Dreyfus in 1980, describes five stages through which a person passes in order to acquire skills needed in a certain area: novice, competence, proficiency, expertise, and mastery [21]. Most people will only reach the "competence" stage in a certain field ([22], p.28). According to the Dreyfus brothers, while one becomes more skilled he or she "depends less on abstract principles and more on concrete experience".
The main characteristics of the stages are:

- **Novice:** Novices need to be given non-situational tasks and a set of rules to fulfill a certain task ([22], p.18f).
- **Competence:** With a certain experience acquired, comes competence. With competence, one can deviate to a certain degree from prior rules given to the novice ([22], p.20).
- **Proficiency:** With proficiency, one can solve known problems, seek guidance from experts and apply the advice given successfully ([22], p.20f).
- **Expertise:** This is the first stage where one is able to reflect and correct oneself. On this level, one oversees the big picture and can learn from experience others made ([22], p.21ff).
- **Mastery:** On this level, one has a huge fund of experience and works best based on intuition. Interestingly, if forced to use rules, persons on the mastery level have proven to become less successful fulfilling their tasks ([22], p.23f).

The Dreyfus model allows us to align the approaches presented below to the different levels of expertise for which they can be used.

3 Case Descriptions

3.1 Characteristics

In this chapter, we report on several cases that were run at universities in Germany and Switzerland and in industrial settings. These cases focus in particular on bringing some of the complexities of real projects into an academic setting, either in real life projects (Chapter 3.2, Chapter 3.3) or using role games (Chapter 3.4). Another case description shows the usage of improvisation theatre in professional education (Chapter 3.5). Additional aspects of these cases that are not directly related to RE are "also discussed elsewhere, [19, 20, 23, 24, 25, 26]".

The case descriptions emphasize various aspects, such as

- Interdisciplinarity and complexity in social interactions, due to different skills and backgrounds of the participants (Chapter 3.2, Chapter 3.3)
- Approximation to reality (Chapter 3.2, Chapter 3.3)
- Methodical rigor (Chapter 3.4) vs. "real world muddling through" (Chapter 3.2, Chapter 3.3)

These cases can be categorized with respect to Dreyfus' levels of competence as indicated in Table 1. The levels were assigned based on asking the participants about

their experiences and our observations during the course. We define a teaching resp. training experience as successful if the training objective is achieved. Empty entries in the table are still open for future research. Table 2 gives a short overview of the cases.

Table 1. Participants´ Dreyfus levels on which the authors applied the training method successfully. The numbers refer to the chapter where the case is presented

	Impro theatre	Role game	Simulation	Real life
Novice	3.5	3.4	3.4	3.2, 3.3
Competence	3.5		3.4	
Proficiency	3.5		3.4	
Expertise	3.5			
Mastery				

Table 2. Overview of the cases presented in this paper

Case	**3.2**	**3.3**	**3.4**	**3.5**
Learning objective	Elicitation and negotiation of requirements, understanding the roles of other stakeholders, communication across disciplinary boundaries	Methods for elicitation, specification, management, soft skills, understanding of the user's role in the process,	Elicitation methods, specification methods, soft skills	Soft skills and their specific aspects in RE-related situation such as requirements clarification, prioritization
Learning method	Real life projects with internal or external stakeholders	Real life projects with external stakeholders	Project simulation including role games	Interactive games from Improvisation Theatre, storytelling
Course Size	25-30	25-40	4-25	???
Group size	10-25, depending on the number of students and customers	5-12, depends on the number of students	2	8-25, depends on trainer's experience
Success criterion	Customer accepts project outcome. Self-reflection on achievements and failures in a post-mortem review.	Projects are conducted in a real life situation. Customer accepts results. Additional written test with reflections on methods.	Requirements specification and test cases satisfy quality criteria, customer accepts prototype	Tasks per games are solved, anticipated results are achieved

Important dimensions for the description of the cases are the following:

- Controllability: The instructor´s ability to control and adapt the initial conditions and the course of the learning experience.
- Co-location: eligibility of the approach to be run with a distributed team or in a co-located fashion.
- Feedback types: moment in time and method used by the instructor to obtain feedback about learning success.
- Supervision need: Need of the learners to be supervised by instructors.
- Requires theoretical / practical knowledge of participants: prerequisite RE knowledge for the course.

These dimensions will be discussed in the chapter 4.

3.2 Joint Project with ICT and Business Students

Description

One approach to gain experiences with some of the complexities of real projects is the students' work in teams on projects that have a realistic goal or even a real customer. In particular if there is a real customer, interdisciplinary aspects come into play since in general the customer is active in a different application domain, i.e. students and customers do not share the same disciplinary background. In the cases that we ran at Coburg, we emphasized this even further since the project teams consisted of ICT and business students who had to establish ways to cooperate even across disciplinary borders in order to succeed in the project. In addition, project teams tended to be fairly large, giving rise to unexpected social interactions and coordination problems [24, 25]. Participants are in the final year of their bachelor's studies. Each of them already passed one semester of compulsory internship.

So far, we ran three iterations of such a project. Project I dealt with developing a software system in order to support claims handling in a (fictitious) insurance company. Business students played the role of the customer, expressing requirements and being involved in acceptance testing, while ICT students were in charge of building the system after figuring out what the system was supposed to do. In contrast to project I, there was a real customer in projects II and III (CEO of a medium-sized factory). Each of the projects ran over a complete term, i.e. roughly four months, calling for a weekly effort of approximately four hours for each participant. Each project was concluded by a post-mortem review which focusses on achievements and failures in the project. Furthermore, instructor observed the participants' behavior during the project.

As learning goals, participants should be capable of eliciting and negotiating requirements across disciplinary boundaries in a co-operative manner. Furthermore, participants should get a deep understanding of the roles of all the involved stakeholders. Learning goals are assumed to be achieved if the project outcome could be happily accepted by the customer and participants appropriately reflect their work in the post-mortem review.

Experiences

Each project fostered a much deeper understanding of the importance of requirements and the difficulties in handling them properly. Although both ICT and business students had been introduced to RE, they still did not really believe that there is a problem. In particular, in project I business students (in the role of customers) initially thought that quite a few things simply go without saying (for example the log on process). They assumed that ICT students would fill in the gaps that, from their point of view, were so evident that they would not bother addressing them explicitly. Conversely, ICT students had not expected that their customers, consciously or not, would tell them only part of the story. This experience for both sides was reinforced during acceptance testing: business students first complained about missing important functionalities of the delivered software product, but had to accept that they never expressed a requirement that mentioned these features. For the ICT students, it was a new experience that there were still hidden requirements, even after having asked their customers several times if there were additional issues that the solution should cover. Similarly, students initially tend to believe that requirements never change. Furthermore, projects can help to understand that other stakeholders may have a different perspective on particular things.

But there are quite a few issues that are hard to handle in projects. First of all, the supervision of projects is difficult for larger numbers of students. As a second difficulty, it is hard to foresee what will happen in a specific project, especially when an external customer is involved. Therefore, it is hard to force particular phenomena, e.g. misunderstandings or requirements changes. Consequently, the learning outcome is to some extent left to chance, namely that a particular phenomenon actually happens in a project setting. If the focus of the learning arrangement is on a particular set of phenomena, other formats, such as role games, are more appropriate than projects simply as they are easier to control, yet at the expense of realism.

3.3 Teaching Requirements Engineering to Business Students

Description

At Hamburg University of Applied Sciences [UAS], we continuously conduct a joint course for marketing bachelor students with marketing and RE content since winter 2009/10 [26]. About 30 marketing students are working every semester in 4-6 real life projects which last 7 weeks. The course is in the last semester, so that all students have business experience of at least 6 months. The aim of the task is to solve a marketing problem with ICT support. The students have to define the requirements and then to decide about a software solution, to improve the usability of a web site, to implement a small solution etc. Participating organizations are commercial organizations as well as departments of the university or non-profit organizations.

One professor for marketing and one for business informatics teach and coach the student groups in project management and RE. In some lessons, both professors stand together in front of the class and demonstrate different professional and individual points of view. For special tasks (usability tests), other departments of the university are co-operating. The students organize this co-operation process themselves.

Intended learning outcomes are (a) methodological knowledge for project management and requirements engineering and (b) "soft skills" from the experience of real life projects.

Experiences

At Hamburg UAS, a periodical evaluation of the courses is implemented. The students' feedback is generally positive; they state learning success a well as fun. The different professional cultures of marketing and business informatics are perceived as a confusing, but realistic impression.

For the participating organizations, these projects are important and the students' expertise is accepted as a professional expertise. This is important for the self-confidence of the students and for their role change as future professionals at the end of their university years. Some organizations conduct several projects consecutively with us, so that a student group will continue the work of a former team. In reality, this is a normal situation, but it is not common in teaching project management.

Students criticize the expenditure of time (which correlates to the number of credit points), but first of all the organizational problems. Most of them are caused by the real life situation: Stakeholders have to react on a shift in priorities etc.

The success rate regarding the students' point of view is 100% - no project work deliverable was rejected by the co-operating organizations. The implementation rate of the projects is > 80%, only few of the projects have not been implemented due to changes in the co-operating organization or in their environment.

Problems of the real life situation are:

- Students have a pressure to succeed – therefore the projects have no "gaming" or "exploring" character.
- Due to the required skills and the current curriculum, it is only possible to run such a course at the end of the BA curriculum.
- Due to the different aims in the projects – from implementing only changes on a web site to developing algorithms fur customer clustering -, the focus on methods is different. This is challenging for the students and the teachers, but demonstrates the context-sensitivity.
- Project work can only indirectly reflect students' success. To guarantee a common basic "body of knowledge", a written exam is the base for the grade. The project work can affect the grade positively.

3.4 Requirements Engineering for Engineers

Description

The following format for a role game project simulation worked well for computer science students as well as for business informatics and electrical engineering students, and also for experienced practitioners: Each student plays the role of a customer who wants to get custom-made software and is the provider/ contractor for "another students' project". They work in pairs and change roles. They choose a project which they have implemented themselves before, a problem they have met or

something they will implement soon. The most frequently chosen (and most simple) example was the design of a web site or web shop. But also more complex projects were chosen like steering a manufacturing system or the watering of a system of tennis courts. The objective of the project is to write a requirement specification, test cases and to develop a graphical user interface prototype. No software implementation is needed. However, students who are very experienced in web programming volunteered to "develop the customer's website as a prototype".

This role game has successfully been applied four times:

- In a lecture for computer science and business informatics students at the Technical University of Braunschweig, Germany (three groups),
- In a lecture for business informatics students at the University of Bern, Switzerland,
- In a summer school for engineering students and practitioners at the University of Stuttgart, Germany, and
- In a summer school for computer science students and practitioners, at the University of Applied Sciences in Furtwangen, Germany.

In the university context, the project simulation took the whole semester and the exercises were partly done as homework and partly during the course. The summer school courses took two whole days and no (or few) homework was possible to be done. So, the exercises were all done during the course and took more than half of the course time. Therefore, a shortened version of the project was executed then, with less software artifacts to be written.

The students were led step by step through the process of requirements elicitation (using interviews, but also creativity techniques), UML specification, prioritization, and the implementation of a prototype and its acceptance test by the customer. Before each activity, the instructor provided theoretical knowledge about how to execute a method and standards of notations. As several teams work in parallel, the trainer cannot supervise all interactions but gets feedback about the learning progress when reviewing intermediate results.

The teaching objective was that the requirements specification satisfies the typical quality criteria (completeness, consistency, etc.) and the customer accepts the prototype.

Experiences

The role games make the course a lively and interactive experience. As the same example project is used consistently from beginning to end of the course, the participants see how different RE methods for elicitation, specification and prioritization work together. Errors made in earlier steps are felt in later steps. However, the learning experience is less easily controllable by the trainer than when executing separate role games.

The participants are highly motivated to do the specification well because they have a customer who is interested in the project, and sometimes the product is even planned to be built. This is more motivating than to describe the same library system as the other students in the same course and in the years before.

The project simulation worked well with novices as well as with advanced participants who have work experience. The novices need more support and direct feedback during the exercises. Different participants learn different lessons from the same experience: Novices learn the RE methods and modeling notations, while advanced participants discuss with the lecturer more advanced questions like the ideal level of granularity or questions from their practical experience.

For the teacher, it is an advantage that the projects are all different. This makes the correction and grading of the specification documents an interesting task. And when working in homework, students ``cannot copy other students´ results".

The role play in this form makes only sense with a maximum of 20 participants. Students are working in two person teams (with one three person team, if the number is odd).

The projects always are very different in complexity. It is important to tell the students that it is more important to apply the methods correctly than to end up with a complete specification. This is the difference between this exercise and a real project.

When the course includes homework, it is important that all homework can be done alone and those exercises which must be done in pairs, take place during the course times to guarantee meetings.

So, all in all, this form of the course demands a constant and individual steering by the trainer, who must be able to understand all projects in the course and help with their specification. This demands more than understanding just one sample solution.

3.5 Using Improvisation Theater to Create Interaction

Description

The REIM format (Requirements Engineering and Improvisation) has been developed to train both factual knowledge and soft skills related to RE [19]. REIM follows a typical Improvisation Theater training session but utilizes Storytelling elements to adjust to the participants' background (see [20] for details).

Each REIM workshop session consists of the three phases: warm-up, training and feedback. During each phase different types of improvisation games are played. Only the training games focus on the training of factual knowledge and soft skill competences [20]. Each training game addresses several related soft skill competences. As REIM is quite flexible in this aspect, the trainer prepares upfront which competences shall be addressed. On the other hand, given sufficient experience, games can be adjusted to the participants' needs as the workshop goes along. (This seems to be quite common in industrial trainings)

For instance, RE prioritization is mainly taught and trained fact-oriented. One learns which methods exist and how to apply them, but the participant is left alone to realize this experience. REIM creates this experience by utilizing the "Requirements Game", which – among other aspects – demonstrates priority setting and its difficulties. This is addressed by bringing the participants into a situation where they are so busy fulfilling the (factual-oriented) task given, that they forget to obey other rules which implicitly undermine the priority setting.

The workshop has proven to work well for novices as well as experts in the field of RE. The Improvisation games being used are the same for both parties. Yet, the stories being told differ. Interestingly, the reported personal learning outcomes differ

depending on the degree of experience starting from the creation of numerous aha-effects for novices reaching to intense discussions among participants for experts.

Experiences

REIM is an interactive format which activates each and every single participant. This is in fact what the warm-up games are used for: They are creating the atmosphere. There is no option of participating by observing.

REIM very quickly connects factual knowledge to soft skills and creates experience for each and every participant.

REIM has been applied to numerous groups and was always well perceived. The level of experience in these groups differed from novice to experts, maybe even mastery. During feedback, it became clear that different participants report on different issues that were most valuable in their learning. This seems to be related to the different stages of the Dreyfus model [21], but has not yet been validated. In addition, each participant rates his personal learning-to-having-fun ratio. Interestingly, more experienced groups tend to report a higher learning experience whereas beginners emphasize on the aspect of having had fun.

REIM can be used for up to 40 participants, working best with 15-25. More participants could be (and have been trained) by splitting up the group into two and using a second trainer or repeating the session. This might however result into different non-comparable learning.

From a trainer's perspective, the REIM format appears to be quite complex. In order to utilize the format, the trainer needs some knowledge in improvisation theater games as well as storytelling. It is the trainer who dominates and steers the approach (and thus the success of the workshop) quite significantly.

Even though a Train-the-Trainer description has been formulated as a pattern and a trial session with other trainers was run [23], this complexity might still prevent the workshop format from becoming more widely spread in the community. This might be particularly true for the university area where trainers are often not trained prior to teaching but rather thrown in at the deep end.

REIM appears to be an interesting workshop format to combine factual knowledge and soft skill training into one. It has however not yet been formally validated, nor has the correlation to the Dreyfus model been proven to be valid. This is part of a current investigation of one of the authors and shall result in a sophisticated understanding who REIM works within different group set-ups.

4 Discussion

The above case descriptions provide only a part of the four authors´ teaching and training experiences. Based on the experiences described in this paper and on additional experience, Table 3 describes the different preconditions for four forms of experience-oriented learning methods.

The table indicates that there is no “silver bullet” and that methods could be and have to be adapted to special circumstances. It must be noted that the positive definition “the x-way of teaching fits to the situation y” does not automatically imply

that x will not fit to any other situation. To evaluate this could be an important task in future research.

We suppose that not only the level in the Dreyfus model but also the concrete situation of the learner will influence the selection of the effective type of teaching.

Table 3. Different preconditions under which four different types of teaching/ training have been used successfully

	Improvisation theatre	**Role games**	**Project simulation, toy project**	**Real-life project with real customer**
Group size	Some games are possible with small groups only and others with large groups	When group size is large, then need to form sub-groups	When group size is large, then need to form sub-groups	Only for small groups, because of limited availability of customer
Controllability	High	High	Average	Low
Distributed team	No	Possible	Possible	Possible
Equipment	Room without chairs	Depends, usually seminar room with chairs	Room with tables, chairs and computers	Work places and meeting room
Supervision need	Active supervision for the whole time	Active supervision for the whole time	Initial explanations, answering questions, solving problems	Regular supervision
Theoretical knowledge of participants	None needed	Must be provided	Must be provided	Must be provided
Practical knowledge of participants	None needed	None needed	None needed, but desirable	Essential for success
Feedback to trainer about learning success	Immediate	Immediate	When reviewing intermediate results	When reviewing intermediate results
Dreyfuss level of participants	All levels	Novice, competence	Novice, competence	All levels. For the levels of expertise and mastery, the character will be more a coaching than a training

5 Conclusion and Future Work

This experience report presents four approaches for teaching RE in academia and training in industry. In particular, we presented findings from four interdisciplinary, game-oriented courses. We analyzed these approaches with respect to different settings in which we applied them successfully and in which they also might be applicable, as well as additional issues that need to be considered when designing didactic settings for different target groups. In particular, we found that each of the techniques we employed is suitable for an audience on the novice level of the Dreyfus model, both for teaching and training. On the higher levels of the Dreyfus model, training on specific topics becomes more relevant. Consequently, methods like role games, project simulation and improvisation theatre are appropriate on these levels, primarily due to their good controllability. It should be noted, however, that our findings are derived after the fact from the specific cases that we explored. It is a matter of future work to analyze to what extent our findings are generalizable.

Our discussion about our experiences led to some further research questions, like: How can we assess the level of expertise of the participants ex ante? And which level does the trainer need to have? How can trainers be trained? Further empirical substantiation of experiences as summarized in Tables 1 and 2 would test whether the training methods can be useful for participants on other Dreyfus levels, too.

References

1. Standish Group: Extreme CHAOS (2001)
2. Lethbridge, T.: A survey of the relevance of computer science and software engineering education. In: llth International Conference on Software Engineering (1998)
3. Cap Gemini: IT-Trends 2013, Berlin (October 2013), http://www.de.capgemini.com/ressourcen/it-trends-studie-2013
4. Foppa, K.: Lernen, Gedächtnis, Verhalten: Ergebnisse und Probleme der Lernpsychologie, 9th edn. Kiepenheuer & Witsch, Cologne (1975)
5. Damian, D., Hadwin, A., Al-Ani, B.: Instructional Design and Assessment Strategies for Teaching Global Software Development: A Framework. In: 28th International Conference on Software Engineering, Shanghai, China, pp. 685–690 (2006)
6. Lübke, D., Schneider, K.: Agile Hour: Teaching XP Skills to Students and IT Professionals. In: Bomarius, F., Komi-Sirviö, S. (eds.) PROFES 2005. LNCS, vol. 3547, pp. 517–529. Springer, Heidelberg (2005)
7. http://c2.com/cgi/wiki?ExtremeHour
8. Yusop, N., Mehboob, Z., Zowghi, D.: The Role of Conducting Stakeholder Meetings in Requirements Engineering Training. In: REET International Workshop on Requirements Engineering Education and Training, pp. 133–139 (2007)
9. Regev, G., Gause, D.C., Wegmann, A.: Requirements Engineering Education in the 21st Century, An Experiential Learning Approach. In: 16th IEEE International Requirements Engineering Conference (2008)
10. Mahaux, M.: Improvisational Theatre: an Approach to Soft Skills for Requirements Engineers. In: REET International Workshop on Requirements Engineering Education and Training, pp. 133–139 (2007)

11. Neville, K., Adam, F.: Integrating Theory and Practice in Education with Business Games. Informing Science, Special Series: Informing Each Other 6 (2003)
12. Zapata, C.M., Awad-Aubad, J.G.: Requirements Game: Teaching Software Project Management. CLEI Electronic Journal 10(1), PAPER 3 (June 2007)
13. Favela, J., Peña-Mora, F.: An experience in collaborative software engineering education. IEEE Software 18(2), 47–53 (2001)
14. Baker, A., Oh Navarro, E., van der Hoek, A.: Problems and Programmers: an educational software engineering card game. In: ICSE 2003 Proceedings of the 25th International Conference on Software Engineering (2003)
15. Carrington, D., Baker, A., van der Hoek, A.: It's All in the Game: Teaching Software Process Concepts. In: 34th ASEE/IEEE Frontiers in Education Conference, Savannah, GA, October 20-23 (2004)
16. Fadali, M.S., Robinson, M., Mcnichols, K.: Teaching Engineering to K-12 Students Using Role Playing Games. In: ASEE Annual Conf., American Society for Engineering Education, Washington, DC (2000)
17. Dawson, R.: Twenty Dirty Tricks to Train Software Engineers. In: 22nd International Conference on Software Engineering (ICSE 2000), Limerick, Ireland (2000)
18. Johnstone, K.: Improvisation und Theater. Alexander Verlag (2010)
19. Hoffmann, A.: REIM - An Improvisation Workshop Format to Train Soft Skill Awareness. In: 5th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). IEEE (2012)
20. Hoffmann, A.: Game Language. In: EuroPLoP 2012: 17th European Conference on Pattern Languages of Programs, Hillside (2012)
21. Dreyfus, S., Dreyfus, H.: A five-stage model of the mental activities involved in direct skill acquisition, No. ORC-80-2. Univ. Berkeley Operations Research Center (1980)
22. Hunt, A.: Pragmatisches Denken und Lernen – Refactor your wetware! Carl Hanser Verlag GmbH & Co. KG (2009)
23. Hoffmann, A.: A Trainer's Guideline to Teaching Soft Skills Using Improvisation Theater - A Workshop Format Exemplified on a Requirements Engineering Game. In: EuroPLoP 2011: 16th European Conference on Pattern Languages of Programs (2011)
24. Landes, D., Pfeiffer, V., Sedelmaier, Y., Mottok, J., Hagel, G.: Learning and Teaching Software Process Models. In: IEEE Global Engineering Education (EDUCON), Marrakesh, Morocco, pp. 1153–1160 (2012)
25. Rausch, P., Landes, D.: "Ihr könnt nur zusammen gewinnen" – Interdisziplinäre Praxisprojekte in der Hochschulausbildung. In: Dorn, K.-H., et al. (eds.) Projekte als Kulturerlebnis, pp. 63–74. Dpunkt Verlag, Heidelberg (2009)
26. Weißbach, R.: Bridging the Communication Gap in Information System Projects. Enabling Non-IT Professionals for the Requirements Engineering Process. In: Kettunen, J., et al. (eds.) Applied Research and Professional Education. First CARPE Networking Conference, p. 259. Turku UAS, Turku (2012), http://julkaisut.turkuamk.fi/isbn9789522162519.pdf