5th International Conference on Higher Education Advances (HEAd'19)
Universitat Politècnica de València, València, 2019
DOI: http://dx.doi.org/10.4995/HEAd19.2019.9355

Approaches to promote self-directed learning in Software Engineering

Kristina M. Schulz

Department of Electrical Engineering and Computer Science, University of Applied Sciences and Arts Coburg, Germany.

Abstract

Software Engineering is a profession which requires expertise, but also underlies fast changes of contents and requirements. To cope with these challenges students in this subject need self-directed learning competencies. This paper explores Just-in-Time Teaching as one exemplary teaching method known to support self-directed learning processes. The focus in this analysis lies on a didactical evaluation of the sensitive use of this method concerning the special target group of software engineering students, the teaching content and material. The aim is to avoid a mere transfer of didactical approaches to students without additional value for the learners. Additional didactical consideration are directed towards (already existing) learning strategies of the students, which are essential for enabling students to efficiently benefit from self-directed learning arrangements. Resulting issues need to be evaluated through a serious data collection and analysis, which will be carried out in the subject of software engineering.

Keywords: *Self-directed learning; software engineering; learning strategies.*

This work is licensed under a Creative Commons License CC BY-NC-ND 4.0

1. Introduction

Software engineering is a profession, which requires expertise, but this alone is not sufficient for the fulfillment of the entire range of tasks. Work processes of highly complex disciplines are not sufficiently standardized and are marked by a high degree of variability. This leads to low planning security and constantly developing goals. Multidisciplinary skills and the ability to acquire new knowledge and skills are necessary to meet the requirements of this professional field. Therefore, teaching in the academic training of software engineering must help students to develop skills and abilities that enable the independent acquisition of knowledge, as the permanent changes lead to continuous learning requirements that software engineers even after completing their study phase have to meet on their own. Until a few years ago, academic training in software engineering was characterized by the mere imparting of expertise. But lately there is an educational change of perspective that focuses on the self-management of students' learning processes. The mere transfer of knowledge only partially meets the requirements for the competence profile to be developed in software engineering. Learning should rather be an active process of confrontation between individuals and their (work-) life, in which knowledge and competencies are actively constructed. For this reason teachers should pursue an enabling didactics that initiates and enables self-directed learning processes (Arnold, 2010).

This paper deals with teaching methods that are perceived as supportive for self-directed learning processes. Nevertheless, this perception needs to be substantiated systematically. In particular, sound evidence is needed if they are actually suitable for all groups of students and if the learners always have the right strategies to deal with self-directed learning. Since the last questions might turn out to be false, there should be a deeper analysis of the teaching methods and conditions, in which self-directed learning processes could be promoted efficiently. The main contribution of this paper is to pinpoint some core aspects that need to be examined in such an analysis. Since so far there is no clear definition of what constitutes self-directed learning, a definition is first made and the requirements necessary for self-directed learning processes are presented.

2. Self-directed

When dealing with the topic of self-directed learning processes, one encounters a variety of word combinations with the prefix “self-” such as self-organized, self-regulated or self-active learning. There are several attempts at definition and differentiation (Gnahs, 2002), but for all the disagreement and blurring of the definition, there seems to be at least the largely common view that there is never fully self-determined or completely other-driven learning, but merely “[…] a continuum between two extreme poles that do not occur in reality […]” (Gnahs & Seidel, 2002). According to Gnahs and Seidel, different courses move between the two extreme poles mentioned (self-determined and other driven learning) and can be assigned on the basis of several aspects related to learning arrangements. There is also a noted development from knowledge transfer didactic towards a didactic of self-directed acquisition of knowledge and competencies. These approaches to a "new learning culture" (Siebert, 2006) are based on a constructivist didactic, which assumes that each person constructs its own reality and consequently generates new knowledge by linking it with previous experience (Siebert, 1998). The direct transfer of knowledge by the teacher to the learners is thus not possible in a linear way (Arnold, 1993). This means that the teacher methodically prepares content and information as part of enabling didactics in such a way that the learners take on more and more of the active part in the teaching-learning-process. A change in roles respectively a "shift from teaching to learning" must take place, moving away from simply imparting knowledge towards the design of subject-oriented learning opportunities and the accompaniment of learning processes (Knight & Wood, 2005). In view of the multitude of heterogeneous definitions of concepts, the article aims to use the term and concept of self- directed learning as a process of active self-development of knowledge and competence by the learner with the help of differentiated reflection processes and the facilitation of learner- centric courses by the teachers

3. Requirements of self-directed learning

The process of learning encompasses a transformation of already existing knowledge to expanded knowledge or new skills. This transformation requires cognitive abilities enabling the absorption and processing of information through social and individual processes. Learners must be able to critically review their knowledge assets and competencies with regard to their learning goals in order to determine the starting point of their necessary learning act in a first step (Konrad & Traub, 2013). This process, in which new informations or abilities must be linked to existing ones, can be supported by learning strategies that require an active use of the new and a reduction to its essential content. Mastering metacognition is important for the successful use of these learning strategies. This involves reflexively dealing with one's own knowledge and competency base and the strategies for deriving necessary learning steps, defining learning goals and paths of action (Konrad & Traub, 2013). In addition to learning strategies, motivation should be regarded as a sufficient condition for knowledge acquisition and as important as cognitive abilities. Cognitive learning strategies are first of all a collection of possible approaches, but motivational circumstances determine whether they are applied (Friedrich, 2000). The motivational conditions are distinguished in personal factors and in conditions that are located in the learning situation and can be motivated both extrinsic and intrinsic. For self-directed learning, the intrinsic motivation expresses in a positive, learning-related self-efficacy conviction with volitional strategies is particularly relevant. In this context, Deci and Ryan define three criteria that influences intrinsic motivation. Felt competence/effectiveness, perceived autonomy/self-determination and perceived social involvement significantly promote intrinsically motivated self-directed learning (Deci & Ryan, 1993). Perceived competence/effectiveness depends e.g. on the right level of structure, suitable work material tailored to students and experiences of success, which can be seen in positive feedback. Transparent communication of the learning goals and a classification of the learning content in the overall context of the course also have a positive effect on the perceived self-efficacy of the students. Social involvement of students is achieved if, e.g. they can actively participate in a course and if there is an open learning atmosphere (Meissner & Stenger, 2014).

4. Self-directed learning with Just-in-Time Teaching (JiTT)

To gain skills in the area of self-directed acquisition of knowledge, teaching methods such as Just-in-Time Teaching (JiTT), Flipped Classroom or the Inverted Classroom Method (ICM)[1] seem suitable, because they are seen as to be supportive for the development of these skills. This paper analyzes JiTT as a representative example from this group of methods which are all known to support self-directed learning processes.

4.1. Just-in-Time Teaching

JiTT was first introduced in the 1990s and it was developed in the following years as a teaching learning concept used primarily in STEM (= science, technology, engineering and mathematics) subjects (Novak & Patterson, 1998). Meanwhile, JiTT is used in various disciplines and subjects (Simkins, 2010), but only recently in the field of computer science (Kamph et al., 2013). Originally created with the intention of increasing students' commitment to their field of study, JiTT should give students the opportunity to learn flexibly and efficiently and to train multiple skills such as the ability to acquire knowledge in a self-determined manner, collegial cooperation and communication skills. To achieve temporal flexibility students are provided online materials in the form of reading tasks, learning videos etc. before the next session. The course instructor then corrects the tasks "just in time", evaluates the solutions and incorporates the results of this evaluation into the next course session. The teacher provides feedback on how to solve individual tasks, reflects overall impressions of students' understanding of these tasks, and may address specific problems or questions. This type of teaching design offers teachers the opportunity to get a regular impression of the level of knowledge and understanding of students regarding the current topic. So in the upcoming course teachers can either repeat difficult topics more deeply or

[1] Since Flipped Classroom and Inverted Classroom are used synonymously, in the remaining part of this paper only the term ICM (Inverted Classroom Method) will be used, concerning both terms.

conclude them relatively quickly if the majority of students have no problems with it. For students there are various advantages for their learning habits and the atmosphere in the course: On the one hand students become more familiar with the independent development of new knowledge through the preparatory texts and accompanying tasks, while at the same time learn to organize their time (Hagel et al., 2013; Novak & Patterson, 1998). In addition, the participants feel perceived by the teachers through the regular feedback and can better assess their respective levels of knowledge. They also take joint responsibility for the course, as their practice tasks serve as the basis for the upcoming course session (Simkins, 2010).

5. Reflections on the use of JiTT in software engineering

5.1. The need for self-directed learning in software engineering

In order to a better understanding of the need for self-directed learning in academic training in software engineering education, it is necessary to look at requirements of this field of work: the development of software systems requires many complex and independently executed steps. Customer requirements must be collected in multidisciplinary communication processes, documented in an appropriate form and implemented. It is important to constantly adapt to the requirements of the professional environment and to act appropriately in it. So flexibility, open-mindness and the ability to constantly explore new domains is necessary and only manageable with self-directed learning competency. In recent years the focus in software engineering education has been increasingly on activating teaching methods, which can support self-directed learning processes. This corresponds to the previously mentioned recognition that software engineering is a very complex field of work and its knowledge base is undergoing constant renewals. These characteristics make self-directed learning processes all the more important. It is fair to assume that methods from the blended learning field which use accompanying digital media, could work as a “door opener” for students in a technical subjects such as software engineering to get their attention and implicitly support their self-directed learning abilities.

5.2. Preconditions to a reasonable implementation of JiTT

Despite all interest in JiTT in recent years, this method must not be adopted and transferred to courses with the intention of making teaching more active and learning processes more self-directed without giving it a deeper thought. Learning objectives must be determined beforehand to characterize precisely what shall be learned in the course, before the question of how the content should be provided can be addressed. The following factors must be taken into account in the didactic design of self-learning environments (Jenert & Zellweger Moser, 2011): Students must already have sufficient learning strategies, e.g. the ability to structure their time or the ability to prioritize the contents that have to be learned. Equally important is the ability to perceive, assess and, if necessary, correct one’s own learning activities and processes in a reflective manner (Jenert & Zellweger Moser, 2011).

These skills cannot be assumed as granted. The two teaching-learning-methods described above do not explain strategies of self-directed learning, but give students the opportunity and the freedom to self- directed learning. This means that without further guidance or prior consideration of the target group for this teaching-learning-method, there is a risk that students will be overstrained. Furthermore, the motivation of the students is relevant, which must not be out of focus, since especially in self-directed learning processes the intrinsic motivation is decisive for the learning success. In addition students should be given enough space for exchange and interaction, since "learning in the constructivist sense is understood as a social process." (Jenert & Zellweger Moser, 2011).

6. Studies on the effect of JiTT in software engineering training

In relation to the teaching method presented, an analysis is currently being carried out at the University of Applied Sciences and Art Coburg on the use of the methods and their didactic background. For several years now, a course in requirements engineering has been offered here, using JiTT. In another basic course ICM is implemented in the form of learning videos provided to students on the Moodle platform. In the university context, there are already some multidisciplinary studies dealing with the use of JiTT and the impact on student satisfaction with the course and how helpful students assess the method in terms of their own learning behavior and a possible improvement in their academic achievements (Meissner & Stenger, 2014). Some studies focus on the teachers' perspective on the use of JiTT and, e.g. on the amount of work done by lecturers (Gavrin, 2006). But are evaluations that focus on the question of students’ satisfaction with a certain teaching method or how much work load this method imposes on a lecturer enough for a deep pedagogical-didactical understanding of these methods and their, arguably, positive effect on students learning strategies? So far, there are no studies in software engineering on the question, which aspects of already existing learning behaviour or knowledge acquisition strategies of the student have an impact on the use of these teaching methods and the quality of these effects. The basic research already mentioned examines the framing conditions for the use of JiTT in connection with learning and teaching software engineering. In this context, questions relating to the value of JiTT for learning success and the improvement of teaching in software engineering will as well be examined as the aspect, under which these didactical methods are especially suitable for certain courses and groups, for certain topics and material. Without this reflection on the motives and effectiveness in the use of JiTT or other blended learning methods, these methods could be simplified into a purely methodological instrument that can be used without appropriate suitability for the respective event and group of students and so quickly could lose their real effectiveness.

7. Conclusion and outlook

Changes in social and economic structures are leading to new forms of learning and working, which requires the individuals`s ability to reflect, acquire and develop knowledge. These self-directed learning processes can be supported by certain teaching methods such as JiTT, as they offer students the possibility of independent time structuring, a selection of suitable learning materials and an exchange with other learners in the courses. Still, it needs to be asked if JiTT and other blended learning methods do really support self-directed learning processes, if this support works for all groups of students, for all courses, for all teaching contents, and if students are already in command of learning strategies that enable them to benefit from self-directed learning arrangements. Against the background of the didactic considerations presented, it is important to repeat that while there are many points of reference between the constructivist approach of pedagogical discipline and self-directed learning arrangements in software engineering, in terms of the discipline of software engineering these need to be adapted to the respective factors of the field and the target group. In order to achieve added value for students in software engineering training at universities and to avoid a mere transfer of didactic approaches to this specific target group there will be a quantitative and qualitative data collection among different courses using JiTT or related teaching methods in software engineering education. This study aims to didactically contribute to the targeted promotion of self-directed learning in software engineering, which will enable students to meet a world of permanent changing challenges with the help of self-directed learning.

Acknowledgment

This work is part of the EVELIN project and funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung) under grants 01PL12022A and 01PL17022A.

References

Arnold, R. (1993). *Natur als Vorbild.* Frankfurt: VAS.

Arnold, R. (Ed.) (2010). *Grundlagen der Berufs- und Erwachsenenbildung: Vol. 35. Ermöglichungsdidaktik: Erwachsenenpädagogische Grundlagen und Erfahrungen* (2nd ed.). Baltmannsweiler: Schneider-Verl. Hohengehren.

Deci, E. L., & Ryan, R. M. (1993). Die Selbstbestimmungstheorie der Motivation und ihre Bedeutung für die Pädagogik. *Zeitschrift Für Pädagogik, 39,* 223–238.

Friedrich, H. F. (2000). *Selbstgesteuertes Lernen – sechs Fragen, sechs Antworten.* Tübingen. Retrieved from Deutsches Institut für Fernstudienforschung an der Universität Tübingen.

website:http://netzwerk.lo-net2.de/lfvt/Fortbildung/Paedagogik/ Selbstgesteuertes%20lernen.pdf

Gavrin, A. (2006). Just-in-Time Teaching. *Metropolitan Universities, 17,* 9–18.

Gnahs, D. (2002). Potentiale und Gefahren des selbstbestimmten Lernens. In P. Faulstich (Ed.), *Praxishandbuch selbstbestimmtes Lernen: Konzepte, Perspektiven und Instrumente für die berufliche Aus- und Weiterbildung* (Vol. 37, pp. 99–107). Weinheim: Juventa.

Gnahs, D., & Seidel, S. (2002). Überblick über selbstbestimmtes Lernen in der Weiterbildung. In P. Faulstich (Ed.), *Praxishandbuch selbstbestimmtes Lernen: Konzepte, Perspektiven und Instrumente für die berufliche Aus- und Weiterbildung* (Vol. 37, pp. 13–24). Weinheim: Juventa.

Hagel, G., Mottok, J., & Müller-Amthor, M. (2013). Drei Feedback-Zyklen in der Software Engineering-Ausbildung durch erweitertes Just-in-Time Teaching. In A. Spillner & H. Lichter (Eds.), *SEUH 213: Software Engineering im Unterricht der Hochschulen 2013* (pp. 17–26).

Jenert, T., & Zellweger Moser, F. (2011). Konsistente Gestaltung von Selbstlernumgebungen. In H. Bachmann (Ed.), *Kompetenzorientierte Hochschullehre: Die Notwendigkeit von Kohärenz zwischen Lernzielen, Prüfungsformen und Lehr-Lern-Methoden* (1st ed., pp. 80–117). Bern: hep.

Kamph, T., Salden, P., Schupp, S., & Kautz, C. (2013). Just-in-Time Teaching für Software Engineering. In A. Spillner & H. Lichter (Eds.), *SEUH 213: Software Engineering im Unterricht der Hochschulen 2013* (pp. 9–15).

Knight, J., & Wood, W. B. (2005). Teaching More by Lecturing Less. *Cell Biology Education, 4,* 298–300.

Konrad, K., & Traub, S. (2013). *Selbstgesteuertes Lernen: Grundwissen und Tipps für die Praxis* (4th rev. and enl. ed.). Baltmannsweiler: Schneider-Verlag Hohengehren.

Meissner, B., & Stenger, H.-J. (2014). Agiles Lernen mit Just in Time Teaching: Adaptive Lehre vor dem Hintergrund von Konstruktivismus und intrinsischer Motivation. In O. Zawacki-Richter, D. Kergel, N. Kleinefeld, P. Muckel, J. Stöter, & K. Brinkmann (Eds.), *Teaching Trends 2014: Offen für neue Wege : Digitale Medien in der Hochschule* (pp. 121–136). Münster, New York: Waxmann.

Novak, G., & Patterson, E. (1998). *Just-in-Time Teaching: Active Learner Pedagogy with WWW.* Cancun, Mexiko.

Siebert, H. (1998). *Konstruktivismus: Konsequenzen für Bildungsmanagement und Seminargestaltung*. Frankfurt/Main: DIE.

Siebert, H. (2006). *Didaktisches Handeln in der Erwachsenenbildung: Didaktik aus konstruktivistischer Sicht* (5th rev. ed.). Augsburg: ZIEL.

Simkins, S. P. (Ed.). (2010). *Just-in-time teaching: Across the disciplines, across the academy* (1st ed.). Sterling, Va: Stylus Pub.