Ensino e Tecnologia em Revista

ISSN: 2594-3901

http://periodicos.utfpr.edu.br/etr

# Possibilities for interdisciplinary software engineering education using problem-based learning (PBL) and its social impacts

**Paulo Roberto Córdova**
paulo.cordova@ifsc.edu.br
orcid.org/0000-0001-7283-3106
Instituto Federal de Santa Catarina (IFSC), Caçador, Santa Catarina, Brasil.

**Joel Haroldo Baade**
baadejoel@gmail.com
orcid.org/0000-0001-7353-6648
Universidade Alto Vale do Rio do Peixe (UNIARP), Caçador, Santa Catarina, Brasil.

## ABSTRACT

Software Engineering is an essentially interdisciplinary area. However, by the fact that this area be defined by the Reference Curriculum of the Brazilian Computer Society (2005) as a curricular framework isolated discipline of the computing undergraduate courses, sometimes its concepts are not properly applied. It is common to see the disciplines that compose the Software Engineering decontextualized and fragmented. The problem about it is that this area is too broad and complex to be summarized into a few disciplines. Given its interdisciplinary nature, it requires an equally interdisciplinary educational process. However, regarding the existing pedagogical practices in computing area, it is not observed much interdisciplinary actions. In this context, this work proposes a systematic literature review and some discussion in order to provide an alternative teaching method using PBL. As results it was observed that this teaching technique has potential to be adopted as an alternative applied to the education of interdisciplinary citizens. Practical researches are in progress, aiming to demonstrate such potential.

**ABSTRACT:** Software Engineering. Interdisciplinarity. Education.

## INTRODUCTION

Software Engineering, as demonstrated throughout this article, is an interdisciplinary area of knowledge by its nature. In this sense, the educational process of those who will act in any area of its scope of coverage must be, equally, interdisciplinary. This is most emphatically observed when it is taken into account the social transformations currently made possible by information and communication technologies.

In this respect, since the products to be produced by people who work in this area tend to influence, or even change the ways in which social interactions happen, attention must be paid to ethical, philosophical and social issues in their formation. The technicist and super-specialized profile, derived from positivist prejudices, which, as says Boaventura de Souza Santos (2008), results in the formation of specialized ignoramuses, no longer meets contemporary needs.

Face of this, the disciplinary paradigm of modern science, reflected in the Reference Curriculum of the Brazilian Computer Society (2005), made the Software Engineering reduced to some disciplines of the Curriculum Framework in the Courses of the Computing area that take it as a reference. This hampers interdisciplinary actions aimed at building a global understanding of this area that, given its complexity, has been transformed into an undergraduate course.

Thus, even though in 2008 the University of Brasilia started to offer the first undergraduate course in Software Engineering in Brazil, the others undergraduate courses in the Computing area still maintains it as fragmented disciplines in the curriculum. This creates the need for interdisciplinary educational actions aimed at promoting an educational process capable of preparing citizens whose interests transcend purely technicist issues.

In this sense, this le proposes an alternative to the teaching-learning process of Software Engineering using Problem-based Learning (PBL), being the result of practices, explored bibliographical researches and documentary analyzes. The bibliographical references used to carry out this research is constituted of National Curricular Guide Lines for higher education in the Computing area, current bibliography in Education and Software Engineering, papers published in periodicals, master thesis and doctoral dissertation, as well as others national and international normative documents.

Thus, in this paper we will try to show, first, the main pedagogical and epistemological implications related to interdisciplinarity in general. In the sequence, we will establish conditions for the understanding of the interdisciplinary character of software engineering. After this, a comprehensive overview of the PBL and its possibilities for interdisciplinary teaching will be offered, as well as a summary of chmitd’s (1983) seven-step technique.

Finally, a discussion will be developed, taking into account the National Curricular Guidelines for the undergraduate courses of the Computing area and the possibilities offered by the PBL for the interdisciplinary teaching of Software Engineering.



Ens. Tecnol. R., Londrina, v. 2, n. 2, p. 249-264, jul./dez. 2018.

## INTERDISCIPLINARITY AND ITS PADAGOGICAL AND EPISTEMOLOGICAL IMPLICATIONS

Interdisciplinarity has been subject of discussion in the academic world since the 1960s, having as one of its milestones, the International Seminar on Pluridisciplinarity and Interdisciplinarity, held in September 1970 at the Université de Nice (SOMMERMAN, 2015). Since then, different approaches and definitions have been attributed to the theme in order to conceptualize and construct approaches capable of overcoming the disciplinary paradigm of the modern science. However, despite of this effort, according to Ivani Fazenda (2012), it is not possible, at least for now, to indicate a universal and consensual definition, capable of clearly conceptualizing what interdisciplinarity is.

However, it can be emphasized that most of the different propositions for the term have in common the idea of interaction between the disciplines. In this sense, it is important to understand, before starting the discussions proposed in this work for interdisciplinarity, the concept of discipline.

In this context, discipline can be defined as synonym of scientific knowledge, and as the result of the atomization of the knowledge into areas of science, influenced, mainly, by the mechanistic view of Descartes (GERHARD; ROCHA FILHO, 2012). However, when talking about discipline using this connotation, it is important to consider it from two approaches, which according to Lück (2012), are: the epistemological approach, which is concerned with the way that knowledge is produced; and the pedagogical approach, whose focus is on the organization of knowledge for teaching.

Therefore, when discussing interdisciplinarity it is important, in some cases, to take into account its epistemological and pedagogical aspects, especially when education processes are the focus of discussion. This is because, in general, the greatest difficulties in relation to interdisciplinary construction are related to issues of epistemological nature, especially with regard to the dialogue between the natural sciences and the social and human sciences. In this regard, according to Claude Raynaut (2015, p.31) "An intellectual gap has been dug progressively between disciplines working on the material dimensions of reality (physics, chemistry, biology, natural sciences, etc.) and human and social sciences”.

In this way, it is assumed that the organization of the interdisciplinary pedagogical work depends largely on previous interdisciplinary epistemological definitions and constructions, especially when talking about teaching-learning process of technology. The following consideration illustrates this: science is accountable for production of knowledge; technology is the practical application of the knowledge produced by science; and teaching is accountable for the replication of already qualified and validated knowledge and techniques. Thus, any pedagogical attempt of interdisciplinary teaching that disregards the epistemological bases of the science and technology in relation to interdisciplinarity runs the risk of being frustrated. It would be like starting a construction by the end of the work.

In this way, the need for an interdisciplinary understanding movement from the scientific-technological bases is perceived. We understand that only in this way it could be possible to implement interdisciplinary actions sufficiently consistent to achieve the effectiveness proposed by interdisciplinary teaching-learning.

## THE INTERDISCIPLINARY CHARACTER OF SOFTWARE ENGINEERING

In order to better understanding the interdisciplinary character of the Software Engineering, it is essential to be clear what it is, what it proposes, what it is composed of, and what the purpose of this knowledge area is. It is equally important to emphasize that the present analysis will be related to the epistemological approach of interdisciplinarity.

In this sense, in order to promote a closer approximation with terminologies and definitions of this technological area, we will present the most accepted definition for Software Engineering, presented in the glossary of IEEE Std 610.12-190, which defines it as: "The application of a systematic, disciplined and quantifiable approach to the development, operation and maintenance of software, that is, the application of engineering concepts to software (IEEE, 1990, p.67).

This definition makes it possible to infer, in a general way, that Software Engineering is an area with a wide interdisciplinary scope. It involves from the conception of the idea of the software product, going through all the phases and necessary processes for its development. In addition, this definition also covers the operation and maintenance of the resulting product, thus covering the entire software product life cycle until decommissioned.

This set of processes and management needs makes it clear the need for interaction between people and technology, constituting a sociotechnical system. In this system knowledge from the applied social sciences, human sciences and exact sciences, should be organized and applied to enable the management of software development projects, people training for the project and for the operation and maintenance of systems, quality management, among others.

Besides, for a better understanding of the scope of the Software Engineering, it is important to mention the definitions stated in the Software Engineering Body of Knowledge (SWEBOK) that serves as an international reference for the disciplines related to Software Engineering. The SWEBOK divides the Software Engineering into fifteen key areas, namely: software requirements, software design, software construction, software testing, software maintenance, software configuration management, software engineering management, software engineering process, software engineering models and methods, software quality, software engineering professional practices, software engineering economics, mathematical fundamentals, and engineering foundations (SWEBOK, 2004).

However, it would not be possible to develop an explanation of each of the above-mentioned areas without becoming too extensive. In addition, the notion that all this range is of the domain and interest of Software Engineering, already makes it possible a clear idea regarding its interdisciplinary character. Despite this, a brief explanation will be made about some key areas that are considered fundamental to show the interdisciplinary scope of the Software Engineering.

Although it seems obvious because it belongs to the exact sciences, Software Engineering, as said in SWEBOK (2004), should concern itself with some of the mathematical foundations that lie behind computational theories. In this respect, we can cite the relational theory of sets that bases the relational logic in database and the basic logic itself, which supports theories related to programming algorithms. These concepts need not be known to users of computer systems, thanks to another mathematical concept called abstraction. However, software engineers cannot ignore them.

Leaving the basic topics and going up at the level of abstraction, another key area of SWEBOK (2004) that deserves attention in this work is that called Software Engineering Processes, which establish, among other definitions, four categories for the processes of Software Engineering, including:

- Primary processes: establishes processes directly related to the development, operation and maintenance of software;
- Support processes: which includes processes supporting the primary processes, such as quality assurance and configuration management, which among other things is concerned with the management of software components and their update versions;
- Organizational Processes: which provides support for Software Engineering, such as training, measurements and metrics, infrastructure management, improvement of organizational processes, among others;
- Processes between projects: establishes processes for management between more than one software projects. They include code reuse, product line management, and more.

This key area, as can be observed, establishes relations with disciplines of Applied Social Sciences, especially with respect to quality management, team and user training, and infrastructure management.

However, even closer relations with other disciplines could be established by looking at, for example, the key area that deals with software engineering management. This requires knowledge about math to prepare estimates, about people management to team mobilization and development, about strategic management to align project objectives to organizational goals, about contract management law, and about other no less important knowledge, but that if they were listed in this paper, they would make it too extensive. These interdisciplinary interactions needs can be evidenced in the SWEBOK (2004) but are clearer in another document that SWEBOK refers to: the Project Management Body of Knowledge (PMBOK), also internationally recognized and maintained by the Project Management Institute (PMI).

Finally, just to demonstrate the relationship between Software Engineering and Human Sciences, something seemingly unlikely, a brief description will be given on the key area called software requirements. This key area describes practices for execution of processes for eliciting requirements for the product. Among other techniques described in SWEBOK (2004) for this activity, it is included: interviews, facilitated meetings and questionnaires. These are some essentially human's procedures that requires intra and interpersonal skills. In this regard, it is important to note that knowledge about human psychology in dealing with software requirements is essential to the work. This becomes clearer by looking at what this means in practice. Thus, according to Kashfi, Nilsson and Feldt:

> [...] more recent research highlights that the users' overall judgment of software is not merely influenced by how they perceive achievement of their goals. The judgment is also influenced by how users perceive satisfaction of their personal needs such as 'being

> stimulated', 'gaining pleasure', or 'feeling connected to their loved ones (SASHFI; NILSSON; FELDT, 2016, p.3).

This shows that knowledge from the human sciences is essential for Software Engineering. More than that, they are an inherent part of it. According to Cardozo and Silva (2014, p.26), "[...] organizational psychology highlights interpersonal relationships as one of the main characteristics of organizational success. Without people, there is no productivity, there are no companies [...] ". As a result, most of the software products produced have the ultimate goal of automating human interactions, such as: organizational processes, financial processes, virtual relationship processes, among others. It is important not to abstract software products from the context where they make sense, that is, from the human interactions. Therefore, knowledge from the human sciences should be considered as part of the formation of the software engineer.

That said, if the interdisciplinary nature of Software Engineering has not been clear considering the knowledge that integrates and interacts with each other to make it viable, using the definition of Carneiro Leão (apud MINAYO, 1994, p.51-52), which considers functionality as the central concept of modern science, it can be said that any technology is essentially interdisciplinary. This idea is clearer in the following quotation:

> The technological essence of modern knowledge emerges as Archimedes' lever that shifts and pushes down the interdisciplinary avalanche of information technology. In the interdisciplinarity of its practices, science is born, not of technique, but of the interdisciplinary essence of technologies (CARNEIRO LEÃO apud MINAYO, 1994).

In this view, interdisciplinarity becomes a consequence of the development of science itself, and intrinsic to it. It is the search for functionality that leads us naturally to unite knowledge from different disciplines to build new disciplines.

Thus, regarding to the concept of functionality of science and its relation to interdisciplinarity, it is considered important to cite another author who strongly influenced the discussions on the subject: The austrian astrophysicist Erich Jantsch. In his article: Inter- and Transdisciplinarity University: A system approach to education and innovation, published in 1970, Jantsch advocates a system that he called science-teaching-innovation, in which disciplines, instead of being organized as an a priori system, should be integrated to serve specific purposes.

In this model, Jantsch (1970) structured the sciences in four levels, being: the empirical level, that has the logic as organizational language and is constituted by the explanatory sciences, like the physics, the mathematics and the biology; the pragmatic level, formed by the technologies of physics, biology and other sciences, having as organizational language the cybernetics; the normative level, whose organizational language is the planning, being composed of macroeconomic sciences and law, for example; and the level of objectives, composed of philosophy, religion and the arts, being the anthropology its organizational language.

Interdisciplinarity, in this model, according to Jantch, would try to extract common languages and principles between the mentioned levels. Thus, Jantsch defines three types of interdisciplinarity: teleological, which seeks relations and common languages between the empirical and the pragmatic levels; normative; which seeks to achieve the same objectives, but between the pragmatic and the normative levels; and the objectivized, which seeks relations and common languages between the normative level and the level of the objectives.

To summarize, Software Engineering arose from a purpose: to organize the processes related to the development, maintenance and operation of software. To make this possible it was necessary the integration of disciplines from the level that Jantsch called empirical - mathematics and the physics - up to the level of the objectives - the human sciences. Therefore, we can say that the interdisciplinary character of Software Engineering is in agreement with both Carneiro Leão and Jantsch. In fact, according to the latter, interactions that exceed more than one level are called transdisciplinary. It could be said then, that Software Engineering is transdisciplinary.

## PROBLEM-BASED LEARNING (PBL)

Among the teaching techniques that can make interdisciplinary teaching-learning processes feasible is the Problem-based Learning (PBL). This technique is characterized, according to Ribeiro (2010, p.13), by "[...] to use real-life problems to stimulate the development of critical thinking and problem-solving skills and the acquisition of fundamental concepts [. ..] ".

According to Borochovicius (2012), the PBL was created to meet the needs of the medical course at McMaster Medical School in Hamilton, around 1965. What motivated its creation was the fact that many doctors were being trained in a model that used to privilege the transmission of concepts, but did not included practices and strategies associated with the application of these concepts to a concrete diagnosis (RIBEIRO, 2010).

Since its inception, PBL has been applied in several countries, including Brazil, both in medical courses and in other areas such as Administration, Physiotherapy and Accounting. Its success is due to the fact of promoting an effective integration between theory and practice. Besides, PBL includes in its essence the need for interdisciplinary interactions to solve the proposed problems.

This teaching technique has as central axes the active search to be carried out by the student and the working tutorial groups. These groups aim at the construction of an active pedagogical relationship, that is, focused on the student, seeking to avoid the traditional model characterized by a strong intervention of the teacher. Thus, the tutorials aim to make the student the protagonist in the construction of knowledge.

Regarding to the practical application of the PBL, a well-known and internationally widespread model in higher education is Schmidt's seven-step proposal, described in his paper Problem-based Learning: rationale and description, published in the journal Medical Education in 1983. In this paper, Schmidt (1983) establishes three fundamental principles for PBL: activation of prior knowledge, encoding specificity and elaboration of knowledge by students.

The activation of prior knowledges concerns the ability to bring up knowledges already acquired by the student, in order to identify their deficiencies and enable the creation of a learning plan based on these deficiencies. In addition, it also enables the codification of the knowledge already acquired and added to the new knowledge for the production of new knowledges. The encoding specificity refers to the approximation between the problems and contents to be worked, and the reality of students, their work environment, social or daily life. Finally, elaboration of knowledge by students is self-explanatory. It consists in the adoption of processes where the student is active in the search for knowledge (SCHMIDT, 1983). These principles must guide PBL activities.

Likewise, it is also important, for a better apprehension about this technique, to briefly describe the seven steps established by Schmidt (1983):

- Clarify terms and concepts: After the tutor, represented by the teacher, present a problem for the tutorial group; this problem should be read with the purpose of clarifying terms and concepts that may be unfamiliar to the students. At this moment, it is very important to reach consensus on the involved terms. It is also at this stage that the first text presented by the tutor are read and the prior knowledges is activated;
- Define the problem: At this stage, the group works together to develop a clear definition of the problem proposed by the tutor. This is one of the most important steps, as problematization is crucial in any research work. It is important that the problem is clear, concise, well delimited and stated so that it can be worked on by students;
- Analyze the problem: After defining the problem, the group should work on the elaboration of solution hypotheses. An excellent technique for developing hypotheses at this stage is brainstorming[2]. In this step, once again the activation of prior knowledges is verified, since students are expected to elaborate hypotheses based on knowledge already acquired about the proposed problem;
- Inventory of Explanations: At this stage, all the knowledge raised so far is filtered and organized systematically. During brainstorming, ideas are encouraged and should not be inhibited by judging or prior filters. All the analysis to carry out the organization of ideas must be make in analysis stage. In the end, one must question whether what has been obtained so far is enough to help solve the central problem;
- Formulate learning objectives: Once the knowledge constructed so far has been organized, it is assumed that it is already possible to identify deficiencies that do not allow progress to solve the problem. These shortcomings must also be systematized and organized in the form of learning objectives. It is in these learning objectives that the team will focus on the next step. It is important that assignments of tasks that meet the study objectives are also performed;
- Research: This is the phase where students will individually undertake those studies assigned to them in the previous step. The tutor may indicate basic or complementary references, organize lectures, among other activities in order to stimulate and enrich the research;
- Synthesize and test the newly acquired information: At this stage, the group finally meets again to present, discuss and debate the new knowledge acquired through research. Synthesis reports, individual and

group evaluations should be produced. At this point, the group should be able to address the proposed problem in more depth (SCHMIDT, 1983).

As presented, PBL enables a dynamic interaction that associates interdisciplinary theory, practice, skills and knowledge. When the focus of a teaching-learning process is on a real problem rather than on disciplinary content, interdisciplinary skills are needed to solve it. The PBL has been successfully applied in undergraduate courses in several areas, and it is believed that its success is due to the fact that the students are involved and responsible as the author of their own learning.

## EDUCATION, INTERDISCIPLINARITY AND AN ALTERNATIVE WITH PBL

The National Curricular Guidelines for the courses of the Computing area establish that the undergraduate courses in this area must implement in their pedagogical projects, among other elements: forms of implementation of interdisciplinarity; forms of integration between theory and practice; and encouraging research as an instrument for teaching and scientific initiation activities (Ministério da Educação, 2016, p. 2). This makes clear, once again, the importance of interdisciplinarity in higher education, now formalized in an official document.

However, the motivation to promote actions aimed at interdisciplinary formation needs to transcend curricular formalities. We should seek, primarily, the construction of a less individualistic and more sustainable society, where at least most citizens could develop their intellectual and critical potential.

Technological transformations have enabled the creation of communication technologies that integrate people from different parts of the world, and to deal with these new technologies, which, among other factors, contribute to the viability of the globalization process, it is necessary to be able to integrate the diverse conceptions and realities. This integration should involve the various disciplines and the possibility of access to research, motivating both, professors and students to seek new knowledge (TERRADAS, 2011, p.96).

Thus, it can be inferred that the formation of super-specialized professionals, devoid however, of what Edgar Morin (2002, p.14) calls relevant knowledge, that enables to apprehending global and fundamental problems, no longer meets contemporary needs. In this scenario, the fragmentation of knowledge areas into isolated disciplines can be cited as one of the barriers to building a broader understanding of a given theme. This fragmentation mitgh limit the teaching-learning process, and according to Gadotti (2000, p.1), "represents an essential issue for scientific progress itself."

Regarding Software Engineering, a knowledge area from which technological transformations that influence the forms of social interaction in many different dimensions are evident, concerns with the educational process must be especially serious. It is now possible to observe that software automation is being implemented in a wide range of contexts, such as: professional, social, affective, commercial, financial and security. And as it depends on the evolution of hardware and software resources, there is still space for even more significant advances in these areas.

Given this, it is evident that the educational process in Software Engineering should promote the formation of professionals whose concerns go beyond the technical specificities, because the result of their work will affect societies and individuals. There are serious ethical, political and social issues involved.

In this context, the Brazilian Computer Society (SBC), despite recognizing the need for interdisciplinary understanding as part of the formation in Software Engineering, stablishes it as an isolated discipline, that is, as a simple component of the curriculum framework (SBC, 2005, p.5-6). The problem is that most of the higher education courses in the Computing area base their curriculum on it. On the other hand, the Ministry of Education, through the resolution CNE / CES 136/2012, creates the Bachelor's degree in Software Engineering, making it clear that reducing this area to a simple and isolated discipline is not compatible with its degree of complexity (DOU, 2012, p. 2).

In addition, Software Engineering is an essentially interdisciplinary area, arising from a need to organize the relevant knowledges to the development, operation and maintenance of software. Hence, reducing it to a discipline implies the loss of a global view of the area.

However, what is observed, in fact, is that it has been a component of the curriculum of higher education in the area of computing, that is, a simply discipline. Sometimes divided into two or three phases. This is the fact and it is necessary to develop means to enable interdisciplinary teaching in this scenario.

Thus, given the possibilities offered by PBL demonstrated it is believed that, if it were well conducted, PBL can help to solve the fragmentation, linearization and abstraction issues inherent to the disciplinary paradigm and, in this case, related to the teaching-learning process of Software Engineering. In addition, it can provide means to promote integration of theory and practice and stimulate research initiation as a form of teaching.

In other words, PBL could help us to meet the requirements cited at the beginning and stated in the National Curricular Guidelines for Computing Undergraduate Courses. Thus, an analysis about each one of those requirements and the possibilities of the PBL for its fulfillment will be shown next.

- Forms of implementation of interdisciplinarity: Starting from a proposition of a real world problem, the PBL naturally overcomes the limitation imposed by the disciplinary paradigm, since the resolution of real problems, once it were properly identified and characterized, presupposes the breaking of the disciplinary barriers, requiring, in general, knowledge from different areas of knowledge and interaction with the problem domain. As Dalben (2013) says, since the planning process of the tutorials it is already possible to establish interdisciplinary relationships on the proposed problem, because it is necessary to consider the connections between the whole and the parts, as well as the coherence and complementarity between the themes. In addition, an overview of what is desired at the end of the process, as well as the notion of the relationships between the disciplines involved in solving the problem, is a crucial exercise for success. Considering this scenario, the problems that the tutor can propose, related to Software Engineering, in particular, will necessarily have an interdisciplinary character, because, when proposing problems that may or should be solved through the application of technology, it should be considered that technology has not an end in itself. In this sense, students will have to interact with the environment and the area in which the problem is inserted, to know their specificities and, from this, to elaborate solutions using technology;

- Forms of integration of theory and practice: As discussed PBL has as one of its principles, the elaboration of knowledge by students. This principle presupposes the student's action in solving the problems proposed. In addition, problems can be proposed in which part of the solution is the execution of practical activities, as Meireles and Bonifácio (2015) describe in their article: "Use of Agile Methods and Problem-Based Learning in Software Engineering Teaching: A Report of Experience", Published in the annals of the XXVI Brazilian Symposium on Informatics in Education. In the article, the authors describe their experience in applying agile methods to software development and PBL in solving problems involving the development of smartphone applications;
- Incentive to research as an instrument for teaching and scientific initiation activities: In addition to the principle mentioned in the previous item, which determines the elaboration of knowledge by students, among the seven steps proposed by Schmitd (1983), is the research stage. At this stage, after planning the study objectives, students should conduct their individual research in order to meet the planning objectives. The tutor, by indicating appropriate material, must guide the research phase, in order to achieve satisfactory results for the resolution of the proposed problem.

Thus, despite the curricular issues and the disciplinary paradigm that make it difficult to implement an essentially interdisciplinary education, it is believed that through adequate teaching techniques such as PBL, it is possible overcome the inherent limitations of the teaching-learning process. However, it is needed people properly prepared to execute educational projects like this. According to to Dalben (2013), mobilizing the students for active and interdiciplinary activities, is one of the greatest challenges of the PBL.

## FINAL CONSIDERATIONS

To conclude, it can be said that PBL has mechanisms and principles that make it a technique with a great potential for the promotion of interdisciplinary teaching. In the educational process of Software Engineering, which is the focus of this paper, in particular, it is believed that this technique can contribute significantly, since observed the responsibilities of teachers and students throughout the process.

Because of its interdisciplinary character, Software Engineering needs alternatives capable of integrating theory and practice, research incentive and interdisciplinarity. It is believed that the same contributions that PBL can bring to Software Engineering may also be evidenced in other areas. In medicine, for example, it has been used successfully since the 1960s, a strong indicator of its effectiveness.

Thus, new researches will be carried out in order to verify if the identified possibilities, in theory, are also confirmed in practice. It is considered, above all, that researches such as this, which seeks to find alternatives for improvement in the teaching-learning process in higher education, has a great importance for the construction of societies, whose subjects are provided with visions and concerns that transcend the purely technical specificities.

# Possibilidades de educação interdisciplinar em engenharia de software usando aprendizado baseado em problemas (PBL) e seus impactos sociais


## RESUMO

Engenharia de Software é uma área essencialmente interdisciplinar. No entanto, pelo fato de essa área ser definida pelo Curriculum de Referência da Sociedade Brasileira de Computação (2005) como uma disciplina isolada da disciplina curricular dos cursos de graduação em computação, às vezes seus conceitos não são devidamente aplicados. É comum ver as disciplinas que compõem a Engenharia de Software descontextualizadas e fragmentadas. O problema é que esta área é muito ampla e complexa para ser resumida em algumas disciplinas. Dada a sua natureza interdisciplinar, requer um processo educacional igualmente interdisciplinar. No entanto, em relação às práticas pedagógicas existentes na área de computação, não se observam muitas ações interdisciplinares. Neste contexto, este trabalho propõe uma revisão sistemática da literatura e algumas discussões para fornecer um método alternativo de ensino usando o PBL. Como resultados observou-se que essa técnica de ensino tem potencial para ser adotada como uma alternativa aplicada à formação de cidadãos interdisciplinares. Pesquisas práticas estão em andamento, com o objetivo de demonstrar esse potencial.



**PALAVRAS-CHAVE:** Engenharia de Software. Interdisciplinaridade. Educação.

## BIBLIOGRAPHIC REFERENCES


BOROCHOVICIUS, E. **Avaliação do Problem-based Learning no Curso de Administração**. 2012, 227f. Dissertação (Mestrado em Educação) - Centro de Ciências Humanas e Sociais Aplicadas, Pontifícia Universidade Católica de Campinas, 2012.

BRASIL. Conselho Nacional de Educação. Diretrizes Curriculares Nacionais para os cursos de graduação em computação. Parecer Homologado CNE/CES, de 03 de março de 2012. Brasília, **Diário Oficial [da] República Federativa do Brasil**, 28 de novembro de 2016, Seção 1, p. 26, 2016.

BRASIL. Conselho Nacional de Educação. Diretrizes Curriculares Nacionais para os cursos de graduação na área da computação. Resolução CNE/CES nº 5, de 16 de novembro de 2016. Brasília, **Diário Oficial [da] República Federativa do Brasil,** 17 de novembro de 2016, Seção 1, p. 22 - 24, 2016.

DALBEN, Â. I.L. O ensino por meio da resolução de problemas. In: VEIGA, I. P. A. **Novas Tramas para as Técnicas de Ensino e Estudo**. Campinas: Papirus, 2013.

DA SILVA, E. F. Relação pedagógica no grupo tutorial: desafios e possibilidades das metodologias participativas (ativas). **Revista Diálogo Educacional**, Curitiba, v. 16, n. 50, p. 1077-1092, out./dez. 2016.

FAZENDA. I. **Interdisciplinaridade:** História, teoria e pesquisa. 18. ed. Campinas: Papirus, 2012.

_________. **Interdisciplinaridade**. São Paulo: UCAM, 2002. Disponível em:<http://www.pucsp.br/gepi/downloads/revistas/revista-6-gepi-abril15.pdf>. Acesso em: 25 maio 2017

FREZATTI, F.; SILVA, C. S. Prática Versus Incerteza: Como gerenciar o estudante nessa tensão na implementação de disciplina sob o prisma do método PBL. **Revista Universo Contábil**, Blumenau, v. 10, n. 1, p. 28-46, jan./mar., 2014.

GADOTTI, M. **Interdisciplinaridade** – atitude e método. 1999. Disponível em:<http://ftp-acd.puc-campinas.edu.br/pub/professores/cchsa/lucianeoliveira/Planejamento%20de%20Ensino/Tema%202%20-%20Interdisciplinaridade/Texto%202%20-%20Interdisciplinaridade.pdf> Acesso em: 12 jan. 2017.

GARCIA, C. G.; SILVA, L. O. A Importância do Relacionamento Interpessoal no Ambiente de Trabalho. **Revista Interbio,** Dourados, v. 8, n. 2, p. 24-34, jul./dez. 2014.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. - IEEE. ISO/IEEE Std. 12207: Systems and software engineering - Software life cycle processes. Montréal: IEEE, 2008.

JANTSCH, E. Inter- and transdisciplinary university: a systems approach to education and innovation. *Policy Sciences,* Amsterdam: **American Elsevier Publishing Company**, v. 1, n. 1, p. 403-428, mar. 1970.

JAPIASSU, H. **Interdisciplinaridade e Patologia do saber**. Rio de Janeiro: Imago, 1976.

KASHFI, P.; NILSSON, A.; FELDT, R. Integrating User eXperience Practices into Software Development Processes: Implications of Subjectivity and Emergent Nature of UX. **CoRR.,** v.1605.03783, 2016. Disponível em: <http://arxiv.org/abs/1605.03783>. Acesso em: 25 set. 2017.

MEIRELES, M. A. C.; BONIFÁCIO, B. A. Uso de Métodos Ágeis e Aprendizagem Baseada em Problema no Ensino de Engenharia de Software: Um Relato de Experiência In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 26., 2015. Alagoas. **Anais...** Alagoas: [s. n], 2015. p. 180-189.

MORIN, E. **Os Sete Saberes necessários à Educação do Futuro**. 2. ed. São Paulo: Cortez, 2002.

PRESSMAN, R. S. **Engenharia de Software: Uma abordagem profissional**. 7. ed. São Paulo: AMGH, 2011.

RAYNAUT, C. Dicotomia entre ser humano e natureza: paradigma fundador do pensamento científico. In: PHILIPI JR, A.; FERNANDES, V. **Práticas Interdisciplinares no Ensino e Pesquisa**. Barueri: Manole, 2015.

RIBEIRO, L. R. C. **Aprendizagem Baseada em Problemas (PBL):** Uma experiência no ensino superior. São Carlos: EdUFSCar, 2010.

SANTOS, B. S. **Um Discurso Sobre as Ciências**. 5. ed. São Paulo: Cortez, 2008.

SCHMIDT, H. G. Problem-based Learning: rationale and description. **Medical Education**, v. 17, p. 11-16, 1983.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Currículo de Referência da SBC para Cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia de Computação. 2005.

TERRADAS, R. D. A Importância da Interdisciplinaridade na Educação. **Revista da Faculdade de Educação**, v. 9, n. 16, p. 95-114, 2011.

Recebido: 22 mar. 2018.
Aprovado: 20 dez. 2018.
DOI: http://dx.doi.org/10.3895/etr.v1n1.8068.
Como citar:
Córdova, P. R.; BAADE, J. H. Possibilities for interdisciplinary software engineering education using problem-based learning (pbl) and its social impacts. **Ens. Tecnol. R.**, Londrina, v. 2, n. 2, p. 249-264, jul./dez. 2018. Disponível em: <https://periodicos.utfpr.edu.br/etr/article/view/8068>. Acesso em: XXX.
Correspondência:
Paulo Roberto Córdova
Rua Bela Vista n. 71, Bairro Vila Santa Catarina, Caçador, Santa Catarina, Brasil.