BENEFITS OF THE PROJECT-BASED LEARNING TO COPE WITH COMPUTER PROGRAMMING EDUCATION: A SYSTEMATIC LITERATURE REVIEW

Alexandre Grotta
Instituto Federal de São Paulo (IFSP); University of São Paulo (USP)

Edmir Parada Vasques Prado
University of São Paulo (USP)

Abstract: Educators have been looking for alternative computer programming teaching methods like Project-Based Learning (PjBL) toward students’ benefits, when compared to traditional teaching methods and their benefits. Given this context, this systematic literature review (SLR) aims to synthesize, categorize and describe the PjBL benefits. We adopted a recognized methodology to filter the initial 457 studies. As results, we found 44 studies that reported the main PjBL benefits to the computer programming students as follows: 1) performance; 2) motivation, engagement or interest; 3) professional aspects; 4) Communication; and other six benefits that were cited twice or more.

Introduction

Computer education programs are fronting recent challenges due to high levels of students’ evasion/drop rates, lack of interest by new entrants, and changes in the manner new generations are prone to learn. Part of these challenges at related to i) computer programming education, given this is a fundamental area of the computer education and ii) the computer programming teaching methods (HEA, 2017; Queirós, 2014; Silva Filho, Motejunas, Hipólito, & Lobo, 2007; Vega, Jiménez, & Villalobos, 2012) at both education levels higher and pre-higher, such as technical high schools, are subjected to these challenges (Romeike & Göttel, 2012).

Project-based Learning (PjBL) has been applied as an alternative to the traditional computer programming teaching method in order to decrease part of these challenges, due to reports of students’ benefits, such as increases of motivation and grades (Dierker, Ward, Alexander, & Donate, 2017; Goulding, 2013; Kastl, Kiesmüller, & Romeike, 2016). Anyhow, we did not find a study that consolidates all these benefits; thus, we conduct a systematic literature review (SLR) to investigate the benefits of PjBL to computer education students. This research gap led to the main SLR question: at higher and pre-higher education, what are the PjBL benefits to computer programming students?

In addition to this section, the second section defines the theoretical bases used by this research; the third section refers to the research methodology. The fourth section reports the results and considers its limitations. The last section refers to the study conclusion.
Theoretical Bases

Computer education might be seen as a continuum or a gradient, in which the teaching methods might be a blend of instructivist and constructivist teaching methods. Instructivist methods such as demonstrations and lectures are still been reported as the most preferred one by educators. On the other hand, there is the arising of constructivist methods like problem-based learning, PjBL, puzzle-based learning, among others. However, educators might not adopt a new teaching method if they did not see clear benefits on it, such as an increase at students’ motivation (Biggs & Tang, 2011; Payne, 2009; Porcaro, 2011).

PjBL might be defined as the theory and the practices towards engaging students into a unique and exploratory context, driven by the resolution of problems (DeFillippi, 2001; Romeike & Göttel, 2012). PjBL is closely related to industry and marketing. Project methods and visions were created besides the educational environment such as the waterfall method. Even further, the agile project methods were created towards software development industry needs. In fact, PjBL shares some characteristic with real-world projects, like increasing the number of communication channels among people that participates on projects. Anyhow, to apply PjBL to computer education might not be a straight and forward process due to many reasons. For instance, it requires alternative evolution instruments and methods in addition to the traditional ones (Grotta & Prado, 2018).

But beyond the mandatory measure name grade (Luckesi, 2008), PjBL educators might observe at their classes auxiliary measure, which we name as benefits. These benefits are highly relevant in the international context due to its potential to predict and influence students’ final behaviors and performance (Boruchovitch, 2008). The benefits might count in favour of the student’s transition from the educational environment to the professional environment (Teixeira, Bardagi, & Hutz, 2007). Even benefits back to their own community might be observed (Grotta & Prado, 2018). In general, there is a relationship among motivation, engagement, and interest at the higher education, thus these factors joined together for final analysis (Boruchovitch, 2008; Stelko-Pereira, Valle, & Williams, 2015).

In light of these fundaments, we choose to research grades benefits and additional benefits as well. We select only relevant international studies which related simultaneously: computer programming education, PjBL and benefits. In order to perform this research, we choose an SLR due to its formality, reliability and repeatability characteristics, given it identifies relevant primary studies, evaluate them and interpret these data by a particular topic
of interest (Felizardo, Nakagawa, Fabbri, & Ferrari, 2017; Kitchenham & Charters, 2007). The detailed SLR process is described in the next section.

**Methodology**

This section presents the SLR methodology and process recommended by Felizardo et al. (2017) split into three different phases: planning, conduction, and reporting phases. Planning and conducting phases are referred by their respective subsections as follows. The reporting phase is described in the next section.

**Planning Phase**

This SLR is considered a cross-disciplinarity topic, given it relates both technical and educational topics, represented by computer programming and the PjBL topics respectively. Given this context, we followed two relevant recommendations found in (Petticrew & Roberts, 2006). First, the research questions were created utilizing PICOC (Population, Intervention, Comparison, Outcome, Context) criteria.

To make the SLR more extensive, we employed four different search engines: two general-purpose and two specific-purpose search engines, as recommended by (Felizardo et al., 2017). As general-purpose search engines, we select Scopus and ISI Web of Science. As specific-purpose search engines, we select ERIC (Education Resources Information Center) and Technology Collection, which regard the education and technological topics respectively. We choose for scope both conference and journal publications from the past five years plus publications, thus starting on Jan 2012 and ending on Aug 2017. Finally, the SLR was planned as six sequential steps as recommend by (Felizardo et al., 2017). The following subsubsections details this criteria choose by this SLR.

**Keywords definition:** The base keywords were found on the two major topics of this SLR: "project based learning" and "programming". The synonyms for each keyword were defined as follows. Regarding "project based learning", the synonyms were "project based teaching", "project based instruction", "project based pedagogy" based on (Hasni et al., 2016) and PjBL. The synonyms for "programming" was "software development" based on (Godse & Godse, 2007; Jain, 2004). Additionally, we limited the language to English only in order to safeguard rigor and repeatability of this SLR by future researchers across the globe, given they might not be familiar with the authors’ mother language.
The exclusion criteria were: E1 – Duplicated studies; E2 – None or superficially relation to PjBL; E3 – None or superficially relation to computer programming; E4/E5 – Study relates to a lower/upper education level, such as Fundamental or Master courses; E6 – Study does not relate technology-oriented courses, such as Accounting majors; E7 – Study relates to curricula or administrative subject; E8 – Study with access denied to the researchers. The inclusion criteria were: a paper that related to both PjBL and computer programming (or software development), to higher or pre-higher education level or both, and it has no exclusion criteria;

String definition: Generic Query String: TITLE-ABSTRACT-OR-KEY ( ("project based learning" OR "project based teaching" OR "project based instruction" OR "project based pedagogy" OR "PjBL") AND ("programming" OR "software development") ) AND (Language = "English") AND (date between "Jan 2012-01" and "Aug 2017-31")

SLR steps definition: Step 1: Remove the filter language or dates from the query string. Run it on each search engines and memo result; Step 2: Run the query string (with filters) on each search engines and export the final results as Bibtex format; Step 3: Import data from step 2 into StArt and remove duplicated studies; Step 4: Remove studies by reading the title, the abstract and the keyword based on the planned exclusion criteria; Step 5: Fully read the article and collect results into StArt extraction form. If an exclusion criterion is found during the reading, stop collect and make it as reject according to its reason; Step 6: Export, classify and write the results.

Conducting Phase

We conduct the SLR from September to November 2017, starting from 457 studies and limited to 44 studies after all filtering process. As represented on table 1, the studies received a label, composed by the letter A plus a sequential number. Studies were ordered by the article title.

Table 1
All researches accepted by this SLR

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Study</th>
<th>Ref.</th>
<th>Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>(Goulding, 2013)</td>
<td>A23</td>
<td>(Richard, Shryock, &amp; Lagoudas, 2012)</td>
</tr>
<tr>
<td>A2</td>
<td>(Liu &amp; Yin, 2013)</td>
<td>A24</td>
<td>(Gestwicki &amp; McNely, 2016)</td>
</tr>
<tr>
<td>A3</td>
<td>(Vega, Jiménez, &amp; Villalobos, 2013)</td>
<td>A25</td>
<td>(Zhang &amp; Liu, 2012)</td>
</tr>
</tbody>
</table>
The studies come from four different educational areas, as follows: Computers Science, Information Systems and related 57% of the studies; Engineering, 30%; pre-higher level, 13%

**Results**

This section describes and analyse the SLR main question in the first subsection and describes its limitation in the second subsection.

**PjBL Benefits**

Table 2 summarizes the SLR main question: at higher and pre-higher education, what are the PjBL benefits to computer programming students? It ranks benefits by total studies.

Table 2

<table>
<thead>
<tr>
<th>Benefits of PjBL to computer education students</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>Ranking</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>#3</td>
</tr>
<tr>
<td>#4</td>
</tr>
<tr>
<td>#5</td>
</tr>
<tr>
<td>#7</td>
</tr>
<tr>
<td>#9</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Legend:** Underline refers to pre-higher education.

The major reported benefit were grades. Given the theoretical bases, this is relevant that formal education considers impacts on grades in order to analyze PjBL benefits. Secondly, the referred studies cited the triad [motivation, engagement, or interest] as one of the major outcomes of PjBL benefits to computer programming. This is reasonable given PjBL tries to engage and thus motivates and increase the students’ interest according to the theoretical bases.

Given the close relation between PjBL and the professional environment, the third benefit related to students’ professional benefits. In this case, clients that represented students’ customers. Similarly, there were reported enhancements on communication, given the many communication channels created by a project, in contrast with a classical instructional class, also reported benefits to the students’ communication.

There is also reports of an enhancement to the teamwork and to the problem identification and resolution, which as also very close and/or leads to other related group benefits such as cooperation, collaboration, trust, and confidence, as well as critical thinking.

**Limitations and Future Studies**

There are two main categories of limitations to this SLR. First, the limitations related to an SLR. It includes limitations of the data quality, in which this work was based on: the limitation of language, in this case, limited to English only; the limitation of time, in this case to the past five years; among other (Felizardo et al., 2017; Kitchenham & Charters, 2007). In order to reduce the impact of these limitations, we followed the SLR literature best practices.
and recommend process as much as possible. Second, limitation on how to standardize the various aspects of benefits outcome from PjBL to computer programming education (Bell, 2016; Sampieri, Collado, & Lucio, 2006). To address these limitations, we created theoretical bases to aggregate benefits into common classes of benefits. Future studies may investigate how to create standards regarding these benefits for instance.

**Conclusion**

The main objective of this work was to identify the major PjBL benefits to computer programming students at both higher and pre-higher levels of education. We conducted an SLR which resulted in 44 relevant studies published between Jan 2012 and Aug 2017. The first main benefits reported were improvements to students’ grades performance. Secondly, students’ motivation – and its related benefits interest and engagement. Other benefits were, according to the most cited ones: (#3) Profession; (#4) Communication (#5) problem identification and resolution and teamwork; (#7) Cooperation; and finally (#9) collaboration, trust and confidence and critical thinking.

This research also indicates that both higher and pre-higher education level have many similarities, including benefits to students’ grades and motivation, among other common positive impacts. These findings may help educators, education leaders and related person to identify the benefits of PjBL to computer programming students, at both higher and pre-higher education levels.

**References**


Kastl, P., Kiesmüller, U., & Romeike, R. (2016). Starting out with projects - Experiences with agile software


