Project-Based Learning for Teaching Software Engineering: An Experience Report

Bernardo J. da S. Vieira[1], Lucas D. S. da Rocha[1], Ceres G. B. Morais[1], Maximiliano A. da S. Lopes[1], Pedro F. R. Neto[1], Raimundo N. B. Neto[1], Gabryella R. Rodrigues[2]

[1] Departamento de Informática, Faculdade de Ciências Exatas e da Terra
Universidade do Estado do Rio Grande do Norte (UERN), Av. Prof. Antônio Campos, S/N, BR 110, Km 48 - Costa e Silva, Mossoró, RN – CEP: 59.610-090

[2]Curso de Telecomunicações, Instituto Federal de Educação, Ciência e Tecnologia do Pará, Av. Almirante Barroso, 1155, Belém, Pará - CEP: 66093-020

{bernardovieira,lucasdenner}@alu.uern.br,
{ceresmorais,maximilianolopes,pedrofernandes,bezerraneto}@uern.br,
gabryella.silva@ifpa.edu.br

Abstract. Introduction*: Software Engineering education requires the development of both technical and interpersonal skills. In this context, Project-Based Learning (PBL) has stood out for increasing student engagement and fostering the practical application of course content.* ***Objective****: Accordingly, this study aims to report on the implementation of PBL in the Software Engineering I course of a Computer Science program at a public university.* ***Methodology****: To this end, an exploratory action research approach was adopted, with data collected through participant observation, questionnaires, and interviews. Students developed software projects for real clients, going through the stages of requirements elicitation, modeling, design, and prototyping.* ***Results****: The data revealed increased student engagement, development of technical and socio-emotional skills, and a positive perception of the methodology used. Thus, this experience highlights the potential of PBL to integrate theory and practice in Software Engineering education.*

1. Introduction

Teaching Software Engineering (SE) in Computer Science programs presents the ongoing challenge of balancing theory and practice, especially regarding the development of technical and socio-emotional skills [Pérez and Rubio 2020]. Although methodologies such as Project-Based Learning (PBL) are widely discussed in the literature, their practical application still shows significant variations depending on the institutional context, student profile, and strategies adopted by instructors.

This article presents an experience report that stands out for the systematic and continuous use of PBL throughout the entire semester of the Software Engineering I discipline, offered in a Computer Science program at the State University of Rio Grande do Norte. The approach was distinctive in that it integrated a real project with an external client from the beginning of the discipline, using biweekly deliverables, templates adapted from industry practices, and public presentation sessions with feedback from students,

DOI: 10.5753/wie.2025.13209

the instructor, and the clients themselves, aiming to prepare students for an increasingly competitive job market [Meireles and Bonifácio 2015].

Another distinguishing aspect was the emphasis on formative follow-up with continuous guidance and revision cycles, fostering a collaborative and active learning environment. The class involved had no prior experience with Software Engineering, which amplified the impact of the methodology on knowledge construction and the development of competencies such as teamwork, communication, organization, and leadership.

By reporting this experience, we aim to contribute to the discussion on how PBL can be strategically structured to enhance student engagement and promote meaningful learning in technical courses. The article is organized as follows: Section 2 presents the theoretical foundation; Section 3 describes the research methodology; Section 4 reports the implemented experience; Section 5 presents the results obtained; and Section 6 provides the final considerations.

2. Theoretical Foundation

This section presents the theoretical foundation that guides the work developed. Accordingly, we discuss the concepts of Software Engineering, Project-Based Learning, and how it can be integrated into SE education.

2.1. Software Engineering

Software Engineering (SE) is an essential area in Computer Science education, as it provides a methodological foundation for the development of reliable and secure systems. It encompasses activities such as requirements analysis, design, implementation, testing, and software maintenance [Pressman 2005].

In addition to technical aspects, SE also involves interpersonal and managerial skills, such as problem-solving, teamwork, and effective communication, as highlighted by [Sommerville 2011]. These skills are fundamental to preparing students for the challenges of software development in professional practice. Therefore, it is relevant to investigate pedagogical approaches that promote active and meaningful learning of these topics, such as Project-Based Learning (PBL).

2.2. Project-Based Learning

Project-Based Learning (PBL) is a pedagogical approach that emphasizes active learning through student participation in complex and challenging projects. It encourages students to explore real and relevant problems, fostering a deeper understanding of the subjects being studied [Blumenfeld et al. 1991]. The approach involves several stages: defining a problem or challenge, conducting research and investigation, developing solutions, and presenting the results.

This methodology is grounded in constructivist learning theories, such as those proposed by [Piaget 1973] and [Vygotsky 1978]. Piaget suggests that knowledge is actively constructed by students as they interact with their environment, while Vygotsky highlights the importance of social interaction and cultural context in the learning process. PBL promotes the development of critical skills such as critical thinking, problem-solving, collaboration, and communication—skills that are essential in the modern world and necessary for software engineers to perform effectively in the job market.

2.3. PBL in Software Engineering Education

Software Engineering (SE) requires the integration of theoretical and practical knowledge. Given its applied nature, Project-Based Learning (PBL) has proven to be effective in allowing students to experience real-world software development scenarios [Helle et al. 2006]. By engaging students in challenging projects, this approach fosters greater engagement, motivation, and the development of interpersonal skills [Thomas 2000]; [Gil 2009].

Recent studies reinforce this perspective in the Brazilian context. [Santiago et al. 2023] implemented a didactic sequence based on PBL in SE disciplines and observed improvements in student learning and motivation. [Carvalho et al. 2022], in turn, analyzed 38 national studies and highlighted the recurrent use of PBL in game development projects, with emphasis on software engineering practices. In remote learning contexts, [Souza et al. 2021] reported positive outcomes from using PBL in online disciplines, noting increased student interaction and engagement. [Damasceno 2022] also reported high levels of satisfaction and perceived effectiveness among students who experienced PBL combined with active learning methodologies, such as flipped classrooms.

In this study, the Software Engineering discipline was conducted with support from PBL, aiming to provide a more engaging experience, with integration between theory and practice and greater student participation. The following sections detail the implementation of this approach.

3. Methodology

Conducting research requires a well-defined methodological plan, as its central purpose is directed toward investigating and potentially solving a specific problem. In this process, the adopted method plays a crucial role by establishing a structured path with systematic procedures that guide both data collection and analysis. As highlighted by [Rudio 1985], research methodology serves as an organizational support, outlining the necessary steps to understand and examine the phenomenon under study.

Given the need to both understand and intervene in the educational context analyzed, this study adopted action research as its methodological approach. This choice is justified by its cyclical, participatory, and transformative nature, allowing the researcher to act as a subject of both practice and investigation. According to [Thiollent 2022], action research is particularly suitable when the goal is not only to solve concrete problems but also to produce knowledge relevant to the studied reality, involving participants in a continuous process of planning, action, observation, and reflection. In the field of education, this articulation between theory and practice is essential to qualify interventions and support the improvement of teaching practices.

The research is qualitative and exploratory in nature, as it sought to understand the experience lived by students during the application of the PBL methodology, without aiming for generalizations, but rather for the interpretation and reflection on the observed phenomenon. Three complementary instruments were used for data collection: participant observation carried out by the instructor throughout the semester; a mixed questionnaire containing both closed and open-ended questions; and semi-structured interviews focused on students' perceptions of the learning process and skills development. The last two instruments were designed based on the methodological guidelines proposed by [Souza et al. 2019], who investigated student perceptions regarding the use of Project-Based Learning in Software Engineering disciplines. The questions were adapted to the context of the discipline and the profile of the participants, enabling data triangulation and supporting a more comprehensive analysis of the methodology's effects in the studied educational setting.

The experience was conducted over the course of one academic semester, in the context of the Software Engineering I discipline, within a Computer Science program at a public institution. From the beginning of the term, students were organized into teams and guided to develop software projects for real clients, following structured stages of the PBL methodology. Proposed activities included scope definition, requirements analysis, modeling, design, and prototyping. Deliverables were submitted biweekly via the institutional platform and discussed in class, promoting opportunities for group presentations, peer exchange, and continuous feedback from the instructor.

For data treatment, the information obtained through observation, questionnaires, and interviews was organized descriptively. Open-ended responses underwent thematic categorization, starting with exploratory reading (open coding) and then structured by organizing categories according to semantic proximity, following an axial coding approach. The process was conducted by a single researcher, who was also the instructor of the discipline. Although independent double coding was not performed, analytical consistency was sought through triangulation between participant observation, questionnaires, and interviews. The closed-ended questions were tabulated using simple descriptive methods and analyzed with an emphasis on identifying general trends and variations in student perceptions. The analysis maintained a formative and reflective character, aiming to understand how students appropriated Software Engineering concepts throughout the project and what impact the PBL approach had on their academic and professional development.

4. PBL in Software Engineering Education: An Experience Report

The following section presents the experience report developed within the context of the Software Engineering I discipline, offered in the Computer Science program of a Brazilian public institution during the 2023.2 academic semester. The focus is on the application of the Project-Based Learning (PBL) methodology as the guiding framework of the discipline. As emphasized by [Mussi et al. 2021], experience reports are a valid form of knowledge production, through the critical description of academic and/or professional experiences. They require not only the narration of events but also their reflective analysis grounded in theoretical foundations. Accordingly, this report aims not only to describe the pedagogical intervention but also to highlight the formative elements and challenges faced during its implementation.

4.1. Discipline Characterization

The Software Engineering I discipline is a mandatory curricular component in the fifth semester of the Computer Science program at the institution where the study was conducted, with a total workload of 60 hours. Classified as a theoretical discipline, its syllabus includes topics such as data analysis, systems design, object-oriented methodologies, modeling tools, and applied case studies [UERN PPC 2021].

The discipline was structured into three thematic units. Unit I addressed software engineering fundamentals, process models, and requirements engineering. Unit II covered object-oriented software analysis and design, with an emphasis on design patterns. Unit III included concepts related to software testing and the unified process. Activities were organized through lectures, practical exercises, and the development of an applied project, conducted according to the principles of the PBL methodology. The discipline's assessment combined theoretical exams and progressive submissions of the project stages.

4.2. Class Characterization

The experience was conducted with a class of 21 students regularly enrolled in the Software Engineering I discipline. During the semester, two students did not complete the discipline, resulting in a dropout rate of approximately 9.5%. In the end, 19 students (90.5% of the class) actively participated in the activities and successfully completed the course.

Although the students had no prior exposure to the specific content of Software Engineering, they had already completed introductory disciplines in object-oriented programming and databases. This prior knowledge contributed positively to the assimilation of the concepts addressed throughout the semester, particularly in relation to system modeling and the development of team-based solutions.

4.3. Characterization of the Adopted Methodology

With the objective of integrating theory and practice and enhancing the teaching and learning process in the Software Engineering I discipline, the Project-Based Learning (PBL) methodology was adopted. To conduct the activities, the class was organized into seven groups: five groups with three members each, one group with four members, and one individual project developed by a student who chose to work independently.

The project began on the very first day of class, immediately after the presentation of the General Course Plan (Plano Geral do Componente Curricular – PGCC), to ensure its integration from the first unit and its continuity throughout the entire semester. On that occasion, students received a document describing the project, including a timeline of the stages, deadlines for deliverables, and general guidelines for development. Figure 1 presents the organization of these stages and their respective activities.

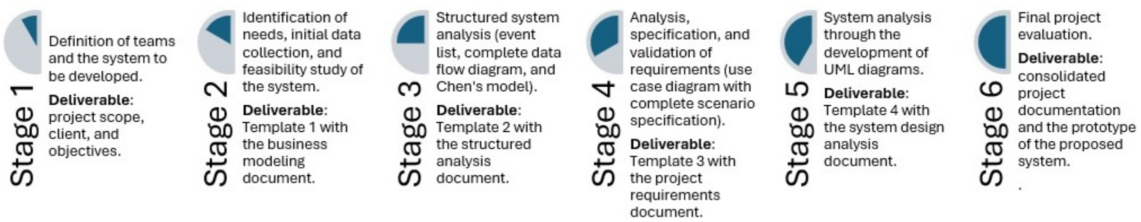


Figure 1. Project Development Stages Throughout the Discipline

Each stage of the project was developed on a biweekly basis, with submissions made through the Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA), the institutional platform used by the University. In addition, the results of each stage were presented orally to the entire class, allowing the sharing of experiences, challenges, and learning among the groups. The discipline's dynamic was structured in an alternating format: one week was dedicated to lecture-based classes focused on the theoretical foundation required for the next project stage; the following week was reserved for group presentations, followed by feedback and pedagogical mediation.

For Stages 2 through 6, students used documentation templates adapted from [Oliveira 2018], based on real-world software development models, which contributed to bringing the academic experience closer to professional practices in the field. The six stages that composed the project are detailed below, organized to integrate theory and practice throughout the semester.

- **Stage 1** In the first stage, students formed their work groups and were encouraged to seek out real clients with concrete needs that could be addressed through software development. Each group was responsible for understanding the client's needs, defining the initial scope, and establishing the project objectives. This approach, aligned with PBL principles, fostered active learning by engaging students in the practical application of theoretical concepts and promoted the development of soft skills, such as teamwork and communication, from the very beginning of the process.
- **Stage 2** After being introduced to the initial concepts of Software Engineering, students deepened their understanding of the client's problem, defined the project objectives, and conducted a feasibility study of the proposed solution. To guide this process, Template 1 was provided, containing fields to formalize information such as project name, team members, client, business environment, and constraints. This stage, aligned with PBL, aimed to stimulate the analysis of real-world problems, introduce business modeling concepts, and develop data-driven decision-making skills.
- **Stage 3** Based on the theoretical content covered in class, students began modeling the systems using concepts such as context diagrams, data flow diagrams, and entity-relationship models. To support this activity, Template 2 was provided, offering guidelines for creating the required modeling artifacts. During class sessions, the instructor used examples drawn from the students' own group projects, which helped clarify the content and encouraged adherence to deadlines. This stage, aligned with PBL, contributed to the development of the ability to visualize the system as an integrated whole and promoted continuous feedback through the biweekly presentations.
- **Stage 4** In this stage, students applied the requirements engineering concepts presented in lectures to perform the analysis, specification, and validation of system requirements. Each group received Template 3, which guided the creation of use case diagrams, scenario descriptions, and a list of validated requirements. In addition, the groups held meetings with their respective clients to ensure the consistency of the information gathered. This stage, aligned with PBL, promoted a deeper understanding of precise requirements definition and the development of communication and negotiation skills, reinforced by the biweekly reviews and direct interaction with users.
- **Stage 5** In the fifth stage, students developed the system analysis and design, applying concepts discussed in class, such as UML diagrams. Using Template 4, each group produced at least three diagrams, including representations such

as state diagrams, sequence diagrams, class diagrams, and interface prototypes. The practical experience allowed students to test and validate concepts prior to implementation, as well as to align system design with the needs identified in earlier stages. This stage, aligned with PBL, reinforced the practical application of technical knowledge and decision-making based on real-world contexts.
- **Stage 6** In the final stage, students consolidated the system documentation and prototype. With the use of Template 5 and extended time for organization, the groups structured and demonstrated the results achieved, seeking client feedback to validate the developed solutions. The final presentations included both the functional prototype and the clients' perceptions of the delivered product. This stage, aligned with PBL, provided essential practical experience for professional development, emphasizing skills such as planning, organization, and responsibility, fundamental aspects for professional performance in the field of Software Engineering.

The project stages enabled students to progressively apply the discipline's content in a practical context. The experience reinforced the integration of theory and practice and contributed to the development of both technical and socio-emotional skills.

5. Results Obtained

The application of the PBL methodology in the Software Engineering I discipline resulted in increased student engagement, as evidenced by the punctuality of submissions, the quality of presentations, and active participation in discussions. In addition to technical knowledge, the development of competencies such as organization, teamwork, communication, and autonomy was also observed. The structured stages allowed students to monitor their progress and continuously apply the concepts discussed in class.

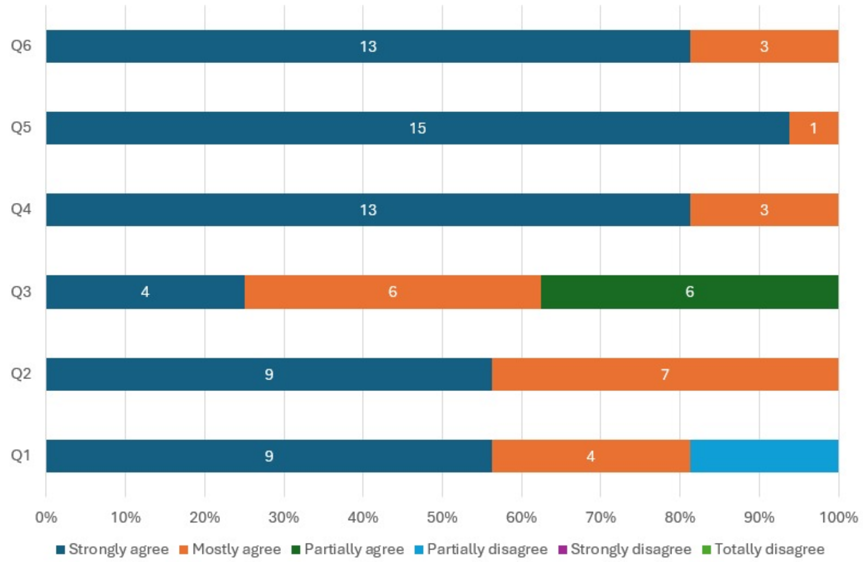
At the end of the semester, 16 out of the 19 students who completed the discipline (84%) responded to an anonymous and voluntary questionnaire. Half of the participants reported having no prior knowledge of Software Engineering, while 25% were familiar only with the theory and 25% had already applied the concepts. Regarding software processes, 50% reported having sought knowledge independently, and 25% had no familiarity with the topic. These data reinforce the importance of pedagogical strategies such as PBL to support students in the early stages of their technical training.

To assess students' perceptions of the discipline, a Likert scale composed of six closed-ended questions was applied, addressing aspects such as motivation, practical applicability of the content, and the contribution of the methodology to professional development. Table 1 presents the wording of the questions used in the instrument.

Table 1. Questions from the Questionnaire on Students' Perception

Questions
Q1. After completing the discipline, do you feel capable of developing a tool/application by applying software engineering concepts and practices?
Q2. Did the methodology used in the discipline add new concepts and practices to your professional development?
Q3. Did you feel motivated during the discipline?
Q4. Did the methodology used in the discipline contribute in any way to your knowledge and training?
Q5. Did the methodology used in the discipline contribute to your academic development?
Q6. Do you believe that the concepts taught in class are important for the job market?

The responses obtained were consolidated in Graph 1, which highlights a predominance of agreement across the different evaluated dimensions. No student expressed strong or total disagreement with any of the statements, which reinforces the positive reception of the adopted approach.



Graph 1: Students' Perception of PBL in the Software Engineering I Discipline

The results consolidated in Graph 1 show that most students expressed strong or full agreement with the statements presented. Questions Q2, Q4, and Q5 stand out, as they relate to the contribution of the methodology to professional development and academic training. In Q1, which assessed the ability to apply knowledge to a real project, nine students fully agreed and three partially agreed, indicating that the experience supported the practical appropriation of the content.

Q3 revealed that, although most students reported high levels of motivation, there was variation in the responses, with some students indicating only partial agreement. In contrast, Q6 showed that the concepts covered were considered relevant to the job market, which enhances the perceived usefulness of the discipline. In the open-ended responses, students highlighted the connection with real-world problems, the clarity of the project stages, and collaborative learning as key strengths. As a point for improvement, some students suggested including more opportunities for theoretical deepening. These findings provide valuable input for final reflections on the impacts of PBL and its potential for improvement in Software Engineering education.

The results presented reinforce the perception that active methodologies, such as PBL, can enrich training in Software Engineering. The following section presents the final considerations of the experience and its implications for computing education.

6. Final Considerations

The reported experience aimed to integrate theory and practice through the application of the PBL methodology in the Software Engineering I discipline, offered in a Computer Science program. The proposal sought to make the learning process more dynamic and meaningful by connecting the discipline's content to real-world project scenarios. The results indicated benefits both in the development of technical skills and in the enhancement of interpersonal competencies such as organization, communication, and teamwork, as well as improvements in socio-emotional aspects, including greater confidence in presentations and enhanced group negotiation abilities—contributing to a more comprehensive education aligned with market demands.

Despite the positive results, the experience also revealed challenges, such as the need to balance lectures and project stages in light of the varying work pace among groups, which required schedule adjustments and additional guidance for some teams. Furthermore, the need for deeper theoretical exploration at certain points became evident, highlighting the importance of seeking a balance between practice and conceptual grounding in future editions. These aspects reinforce the need for continuous instructional support and periodic review of the adopted methodology. It is also important to note that the results observed refer to a single class with a small number of participants, which limits generalizability but does not invalidate the evidence of impact within the analyzed context.

Considering the results, the approach demonstrated potential for replication in other computing disciplines, especially those related to system development and collaborative projects. Future investigations may explore the impact of PBL at different educational levels or combine it with other active learning methodologies, expanding the understanding of its contributions to Software Engineering education and related fields.

References

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., and Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational psychologist*, 26(3-4):369–398.

Carvalho, W. R., Rodriguez, C. L., and Rocha, R. V. (2022). Aprendizagem baseada em projetos no contexto do desenvolvimento de jogos: uma revisao sistemática de literatura. *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 267–277.

Damasceno, A. (2022). Engenharia de software com metodologias ativas no ensino remoto: eficácia percebida e satisfação do aluno em foco. In *Workshop de Informática na Escola (WIE)*, pages 122–131. SBC.

Gil, A. C. (2009). Como elaborar projetos de pesquisa. 12. reimpr. *São Paulo: Atlas*, 6(1-1).

Helle, L., Tynjälä, P., and Olkinuora, E. (2006). Project-based learning in post-secondary education–theory, practice and rubber sling shots. *Higher education*, 51:287–314.

Meireles, M. C. and Bonifácio, B. (2015). Uso de métodos ágeis e aprendizagem baseada em problema no ensino de engenharia de software: Um relato de experiência. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 180.

Mussi, R. F. d. F., Flores, F. F., and Almeida, C. B. d. (2021). Pressupostos para a elaboração de relato de experiência como conhecimento científico. *Revista práxis educacional*, 17(48):60–77.

Oliveira, R. C. (2018). Notas de aulas. `https://www.facom.ufu.br/~ronaldooliveira/aps-2018-2.html`. Acesso em: 20 de março de 2025.

Pérez, B. and Rubio, Á. L. (2020). A project-based learning approach for enhancing learning skills and motivation in software engineering. In *Proceedings of the 51st ACM technical symposium on computer science education*, pages 309–315.

Piaget, J. (1973). To understand is to invent: The future of education.

Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave macmillan.

Rudio, F. V. (1985). Introdução ao projeto de pesquisa científica. *Editora Vozes Ltda*.

Santiago, C. P., Menezes, J. W. M., and de Aquino, F. J. A. (2023). Proposta e avaliação de uma metodologia de aprendizagem baseada em projetos em disciplinas de engenharia de software através de uma sequência didática. *Revista Brasileira de Informática na Educação*, 31:31–59.

Sommerville, I. (2011). Software engineering (ed.). *America: Pearson Education Inc*.

Souza, M., Moreira, R., and Figueiredo, E. (2019). Students perception on the use of project-based learning in software engineering education. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, pages 537–546.

Souza, S. R., Neto, J. C., Paschoal, L. N., and Hernandes, E. (2021). Ensino remoto emergencial de engenharia de software com pbl: um relato de experiência. In *Workshop sobre Educação em Computação (WEI)*, pages 31–40. SBC.

Thiollent, M. (2022). *Metodologia da pesquisa-ação*. Cortez editora.

Thomas, J. W. (2000). A review of research on project-based learning.

UERN PPC (2021). Resolução nº 055/2021 – consepe – aprova o projeto pedagógico do curso de graduação em ciência da computação – bacharelado – presencial – campus central. `https://www.uern.br/controledepaginas/documentos-legislacao-consepe/arquivos/5105resolucao_` n0_2021_055___consepe___aprova_ppc_graduacao_ciencia_computacao___bacharelado___presencial___vinculado_campus_central___anexo.pdf. Acesso em: março de 2024.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*, volume 86. Harvard university press.