SOFTWARE ARCHITECTURE TEACHING
A Systematic Review

Ugyen Lhatshok, Pema Gyamtsho, Thinley Dema, Thinley Rabgay, Zhijia Ren

UNIVERSITY OF CANBERRA

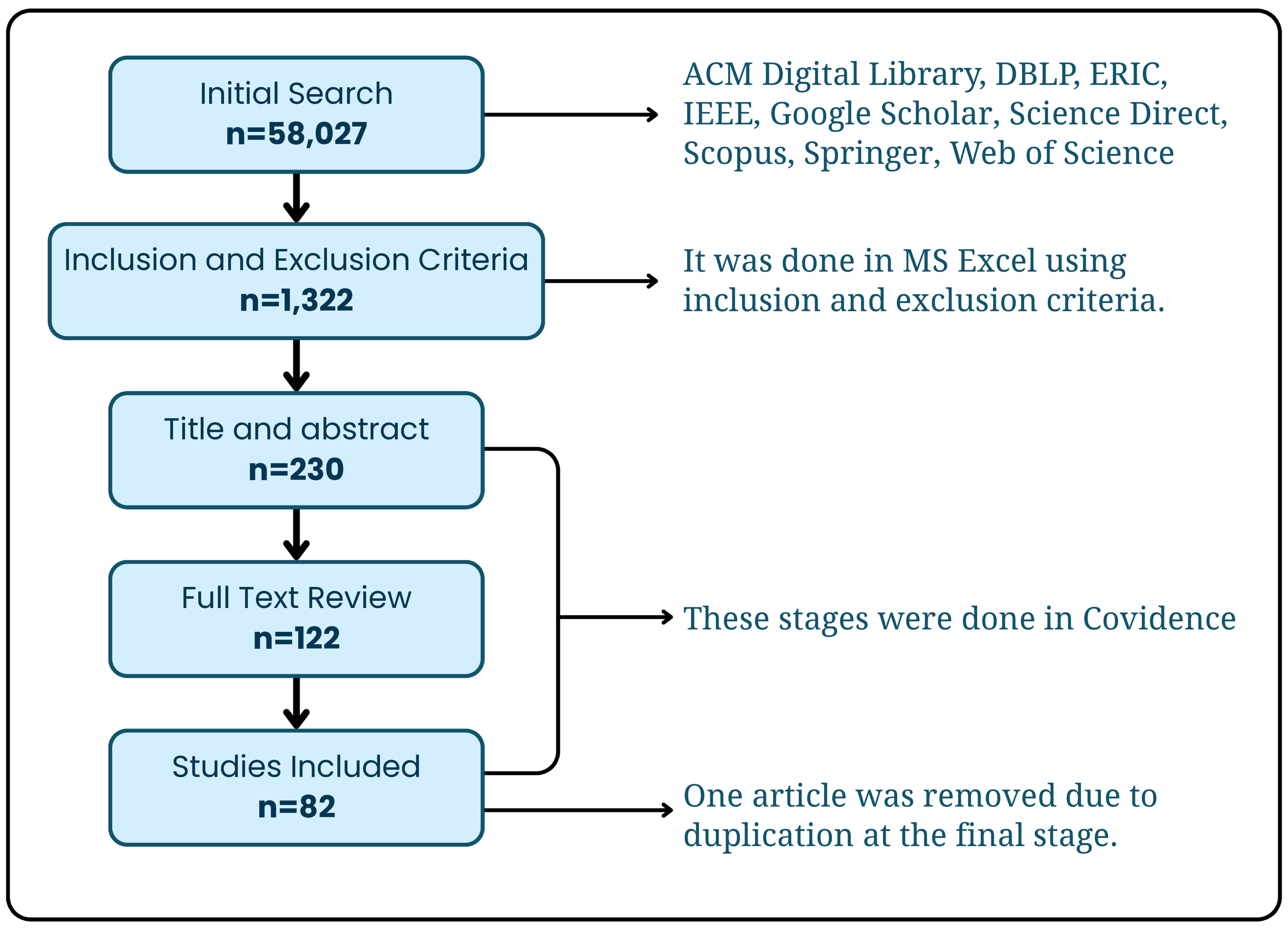
01 Abstract

Software Architecture (SA) education is critical in equipping students with complex software design skills required by the industry. However, SA teaching faces distinct challenges due to its abstract and conceptual nature, high-level thinking and critical reasoning requirements, and the mismatch between SA education and industrial needs. This study analyses 82 peer-reviewed studies from 2010 to 2024, identifying and examining current teaching strategies, their assessment methods, learning outcomes and challenges. This review aims to help educators, curriculum designers, and industry stakeholders enhance the relevance, engagement, and effectiveness of SA education.

02 Introduction

Software Architecture (SA) is important in shaping system structure and performance. Unlike other software engineering topics focused on implementation, SA demands conceptual thinking and design justification. Teaching SA is complex due to its abstract nature and the gap between academia and industry. This review explores teaching methods, learning outcomes, and student challenges, particularly from the students' perspective, providing an integrated view of SA education.

03 Methodology



The SLR follows the guidelines proposed by Kitchenham and Charters: Planning, Execution and Review [1].

04 Results and Analysis

RQ1: Teaching Strategy

10 Strategies were identified

RQ2: Hands-on Experience

Project-based and real-world tasks most prevalent

RQ3: Assessment Methods

Summative and formative methods dominate, but the experiential method is impactful with software architecture, while Kata is underused.

RQ4: Learning Outcomes

Most outcomes fall into mid-levels of Bloom's Taxonomy [2]. But Game-based learning and Problem-based Learning contribute to higher-order level of learning outcomes.

RQ5: Challenges

Abstractness, time constraints, instructor expertise and tool complexity are topmost challenges but can be reduced by teaching suitable theme.

05 Key Findings

- Active learning strategies are more effective than traditional methods
- Project-based and Game-based learning enhance industry alignment and high-order thinking.
- Bloom's Taxonomy shows a gap in higher order skill development.
- Assessment practices need innovation; integrating experiential methods is vital.
- Challenges are interlinked, requiring holistic, adaptive solutions.

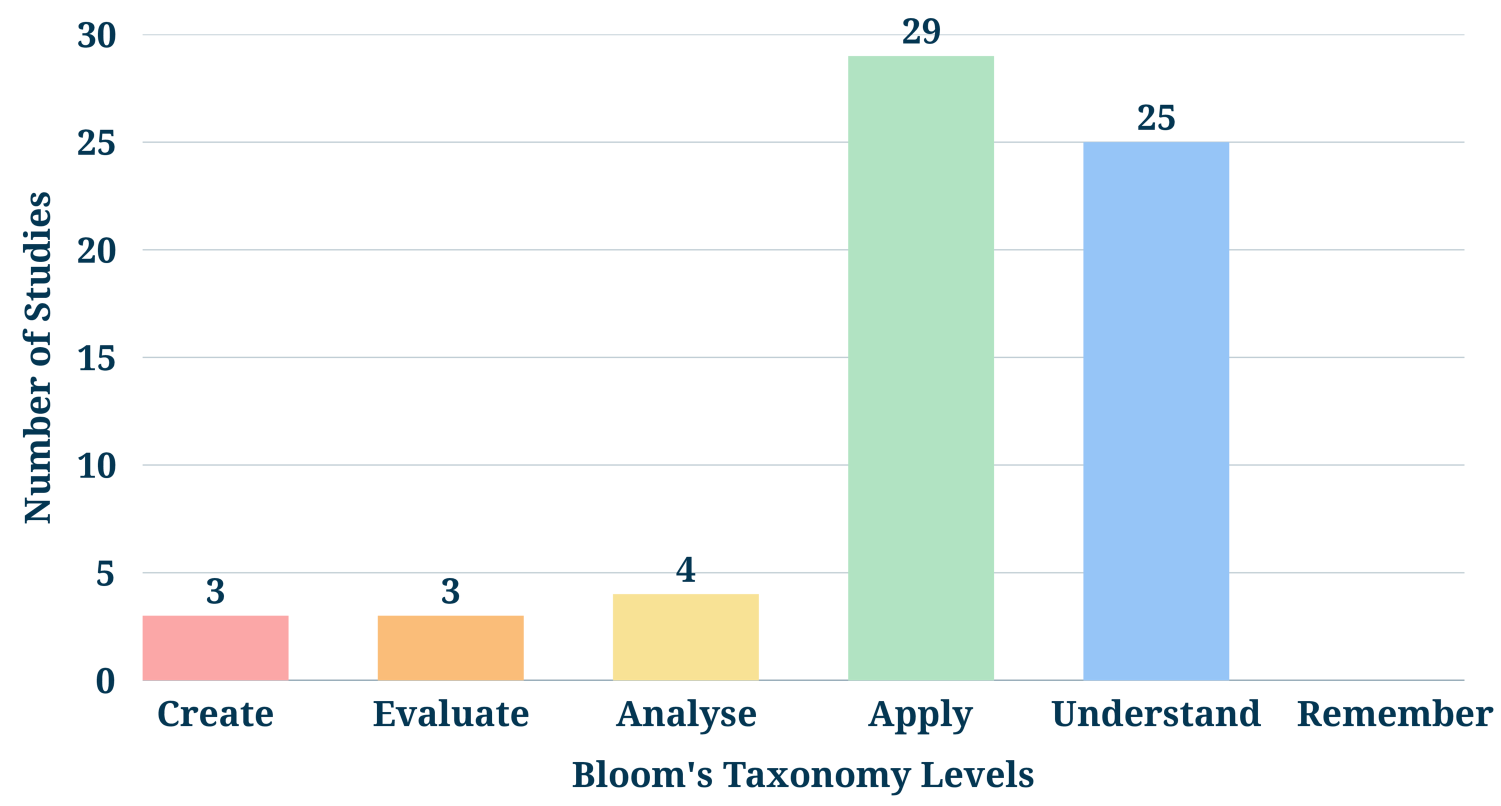


Figure 1: Number of studies aligned in each level of Bloom's Taxonomy [2]

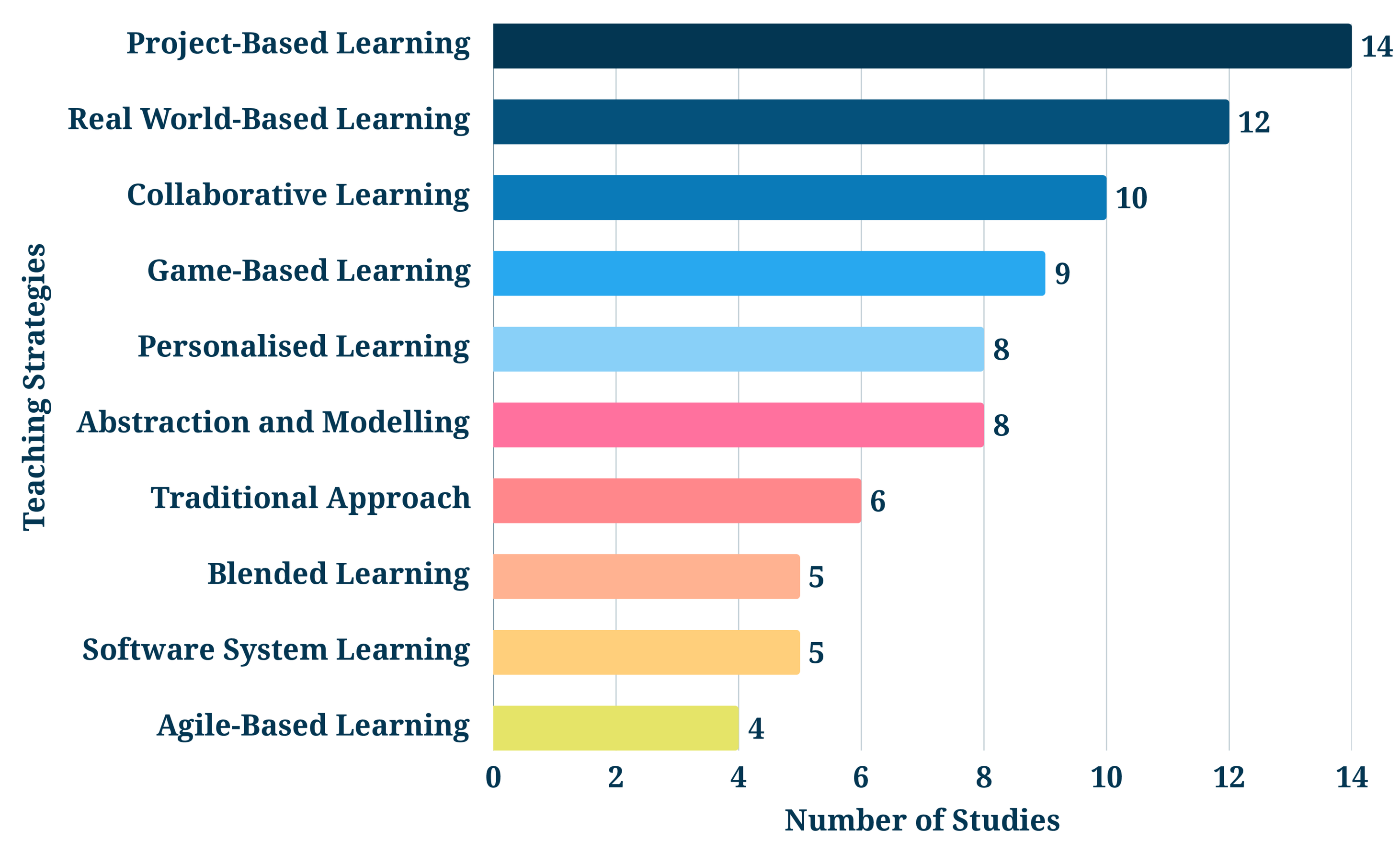


Figure 2: Number of studies categorised under 10 different teaching strategies.

06 Conclusion

This study aims to bridge the academic-industry gap in SA education by adopting blended strategies of PBL and experiential assessment methods. Future efforts should also focus on agile methodology and Architectural Kata to strengthen the effectiveness of SA education and student preparedness.

Acknowledgement

The authors thank the University of Canberra and Unit Convenors for this opportunity to do a systematic review. We are also grateful to our Sponsor and Mentor for their helpful guidance and direction throughout this project.

References

[1] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," vol. 2, 01/01 2007.

[2] N. E. Adams, "Bloom's taxonomy of cognitive learning objectives," (in eng), J Med Libr Assoc, vol. 103, no. 3, pp. 152-3, Jul 2015, doi: 10.3163/1536-5050.103.3.010.