|  |  |  |
| --- | --- | --- |
| **Teaching Strategy** | **Explanation** | **References** |
| **Intuition Driven Design** | Focuses on nurturing architectural intuition through iterative feedback, reflection, and team-based design studio models. | B01 |
| **Experience-based teaching (project-based)** | Uses classroom experiences and projects to share insights into SA education practices (not a prescriptive method, but reflective). | B02 |
| **Board-Based Curriculum (End to End Project-based Learning)** | It uses the full software engineering lifecycle to contextualise architectural decisions and improve holistic understanding in the context of real systems. | B03, B04, B11 |
| **Platform-based lightweight projects** | Engages students with cloud-based platforms (Google App Engine, IBM Cloud Lite) to implement small to medium-sized projects, enabling practical learning and early accumulation of software architecture experience. | B04 |
| **Structured Documentation Practices** | Incorporates architectural documentation as a teaching mechanism to help students articulate and understand architectural rationale and structure. | B05 |
| **Concept-Focused Pedagogical Strategy** | Adapts teaching practices based on analysis of core conceptual challenges and misunderstandings in SA education to improve learning effectiveness. | B06 |
| **Project-based simulation** | Applies simulation through game projects to mimic real-world architecture development and decision-making. | B07, B10 |
| **Model-Driven Teaching (case study)** | Aligns modelling instruction with real-world business needs and professional practices in architectural modelling. | B08, B12 |
| **Scaffolded RE to SA learning** | Students are scaffolded step-by-step from requirements engineering to architectural design in large-scale, distributed system contexts. | B09 |
| **Game based Learning** | Uses educational games to simulate architecture design decisions in a motivating and controlled environment. | B10, B7 |
| **Microservices-based problem-solving** | Engages students in collaborative problem-solving using microservices architecture to reflect industry practice. | B11 |
| **Model driven engineering (MDE)** | Emphasises the use of MDE for teaching abstract system representation and transformation logic. | B12, B8 |
| **Queueing modelling Approach** | Applies performance modelling via queueing theory to teach architectural evaluation techniques. | B13 |
| **Design Studio Pedagogy** | Promotes learning via design critiques, iteration, and minimal lecture time, aligning with architectural studio instruction. | B14 |
| **Ranking Task Activity (Active Learning)** | Students perform prioritization and reflection tasks to build understanding of trade-offs in SA decisions. | B15 |

|  |  |  |
| --- | --- | --- |
| **Teaching Strategy** | **Industry Alignment** | **References** |
| **Intuition Driven Design** | Mimics real-world architecture work where intuition and experiential knowledge drive decision-making in team settings. | B01 |
| **Experience-Based Teaching** | Reflects traditional teaching methods based on classroom experience. It does not incorporate current industry practices or technologies, limiting its alignment with practical expectations. | B02 |
| **End-to-End Project** | Reflects real software lifecycle exposure (from requirements to deployment), aligning with how architecture unfolds in the industry. | B02, B03, B04, B11 |
| **Platform-based lightweight projects** | Familiarises students with modern cloud tools and distributed design challenges faced by architects. | B04 |
| **Structured Documentation Practices** | Develops the habit of structured communication: a key skill in architecture roles across software firms. | B05 |
| **Concept-Focused Pedagogical Strategy** | Identifies critical mismatches between academic methods and industry needs, emphasising the urgency to adopt practical, aligned strategies. | B06 |
| **Project-based simulation  (Game projects and design games)** | Encourages teamwork, design negotiation, and iterative decision-making akin to real SA teams. | B07, B10 |
| **Model Driven Teaching** | Aligns with domains like embedded or enterprise systems where architecture relies heavily on model transformation and formal design. | B08, B12 |
| **Scaffolded RE to SA learning** | Prepares students to handle ambiguity and complexity in distributed systems by guiding them through a structured, industry-aligned pipeline. | B09 |
| **Game based Learning** | Simulates complex architectural decisions through gamification. This prepares students for the cognitive and collaborative processes used in actual architectural roles. | B10 |
| **Microservices-based problem-solving** | Microservices are widely used in industry. This strategy aligns strongly by immersing students in team-based, modern architectural environments. | B11 |
| **Model driven engineering (MDE)** | MDE is used in industry, especially in domains requiring abstraction and automation. However, it's not mainstream across all sectors, giving it moderate alignment. | B12 |
| **Queueing Models** | Emphasises architectural performance reasoning and resource planning: critical in large-scale system design. | B13 |
| **Design Studio Pedagogy** | Mimics architecture firm culture by fostering design critique, iteration, and team-based learning: key to reflective architectural practice. | B14 |
| **Ranking-based Task Design** | Promotes prioritisation and trade-off analysis, mirroring early-stage SA discussions in industry settings. | B15 |

**RQ2: How are students provided with hands-on practical experiences related to software architecture teaching?**

|  |  |  |
| --- | --- | --- |
| **Teaching Strategy** | **Hands on Practical Experiences** | **References** |
| **Intuition Driven Design** | Offers in-class design exercises to help students internalize architectural concepts, but lacks direct hands-on or tool-based implementation. | B01 |
| **Experience-Based Teaching** | Based on lecturer reflections; no structured hands-on activities or student engagement with tools/platforms reported. | B02 |
| **End-to-End Lifecycle Teaching** | Involves full-cycle project scenarios where students follow software development stages, including architecture design and integration. | B03 |
| **Platform-based lightweight projects** | Uses cloud-based tools (PaaS) for student projects, offering realistic environments for hands-on practice with system design and deployment. | B04 |
| **Documentation-Integrated Instruction** | Students engage in preparing architectural documentation (e.g., design reports), providing a limited form of hands-on analytical and communicative practice. | B05 |
| **Concept-Focused Pedagogical Strategy** | Not oriented toward hands-on activities; focuses on understanding teaching challenges, not implementing or practicing SA directly. | B06 |
| **Project-based simulation** | Uses simulated project environments where students actively make architectural decisions within realistic scenarios. | B07 |
| **Industry-oriented modelling practice** | Students apply business-aligned modelling tasks, using industry-informed formats and standards for architecture representation. | B08 |
| **Scaffolded RE to SA learning** | Provides structured hands-on exercises moving from requirements analysis to architecture design, guiding students step by step. | B09 |
| **Game based Learning** | Students participate in role-based decision-making through games, offering interactive and experiential learning in architecture trade-offs. | B10 |
| **Microservices-based problem-solving** | Students collaboratively build real-world systems using microservices, applying architecture decisions in a hands-on software development context. | B11 |
| **Model driven engineering (MDE)** | Involves tool-supported system modelling and transformation exercises, requiring students to create, analyze, and evolve architectural models. | B12 |
| **Queueing theory for performance modelling** | Engages students in mathematical modelling and performance simulation to evaluate architectural decisions. | B13 |
| **Instructional Heuristic in SA** | Primarily pedagogical in nature; may suggest instructional improvements but does not directly involve students in applied activities. | B14 |
| **Ranking-based Task Design** | Introduces structured exercises where students compare and rank architecture options, promoting analytical, hands-on evaluation of trade-offs. | B15 |

**. RQ4: What are the outcomes of software architecture teaching strategies?**

|  |  |  |
| --- | --- | --- |
| **Teaching Strategy** | **Outcome** | **References** |
| **Design Intuition Development** | Helped students internalize core design principles and improved their ability to reason through design trade-offs. | B01 |
| **Experience-Based Teaching** | Offered insights into practical teaching delivery but showed limited evidence of improved student learning outcomes. | B02 |
| **End-to-End Lifecycle Teaching** | Enhanced students' holistic understanding of software architecture across the SDLC, improving their real-world readiness. | B03 |
| **Platform-based lightweight projects** | Strengthened student engagement and technical confidence by applying architecture concepts using cloud-based tools. | B04 |
| **Documentation-Integrated Instruction** | Improved students’ ability to structure and communicate architectural decisions effectively through documentation. | B05 |
| **Concept-Focused Pedagogical Strategy** | Raised awareness of learning gaps in students; indirectly informed the redesign of teaching methods. | B06 |
| **Project-based simulation** | Encouraged deeper architectural reasoning and active participation; improved student confidence in architectural decisions. | B07 |
| **Industry-oriented modelling practice** | Helped students apply modelling methods aligned with real-world expectations, increasing relevance and skill transferability. | B08 |
| **Scaffolded RE to SA learning** | Supported stepwise learning and improved students’ ability to transition from requirements to high-level design. | B09 |
| **Game based Learning** | Boosted student motivation and understanding of architectural trade-offs through active, engaging gameplay. | B10 |
| **Microservices-based problem-solving** | Fostered collaboration and exposed students to realistic architectural patterns, increasing job-readiness | B11 |
| **Model driven engineering (MDE)** | Developed formal reasoning and system abstraction skills, with benefits for students pursuing advanced design roles. | B12 |
| **Queueing theory for performance modelling** | Improved students’ capacity to evaluate performance early in design, though may not be broadly applied. | B13 |
| **Instructional Heuristic in SA** | Provided teaching heuristics that clarified architectural concepts, potentially improving learning outcomes indirectly | B14 |
| **Ranking-based Task Design** | Improved analytical thinking and decision-making through comparison of architectural alternatives. | B15 |

In some ways, it has to discuss software architecture. If not, it should be included in the discussion, the limitation, or the introduction. (Establish how the work differs from the other—in work-related terms.) Some papers talk about training, whereas ours are based on teaching. Identify the unique point from other papers. (specific to the software Architecture) (Critical analysis). Question 4- outcome for the report that we are doing and the survey of 83 papers (results) Teaching strategies should have the results and should be reported (we can discuss this by refereeing the papers in question 1). Result section (effectiveness of the strategies) Challenges about applying the strategies or tools (whatever they have mentioned).

**RQ5: What challenges do students and educators face with the software architecture teaching approach?**

|  |  |  |
| --- | --- | --- |
| **Teaching Strategy** | **Industry alignment** | **References** |
| **Design Intuition Development** | * Difficulty in teaching architectural thinking without extensive project experience. * Students struggle to internalise design trade-offs in abstract scenarios. | B01 |
| **Experience-Based Teaching** | * Lack of structured materials and standardisation in SA instruction. * Students often miss the broader context of architecture beyond code. | B02 |
| **End-to-End Lifecycle Teaching** | * Requires instructors with diverse industry and academic experience. * Balancing broad curriculum content (business to deployment) within limited time is challenging. | B03 |
| **Platform-based lightweight projects** | * Access to and management of real-world platforms (PaaS) can create technical overhead. * Students may struggle without prior cloud experience. | B04 |
| **Documentation-Integrated Instruction** | * Students underestimate the value of documentation. * Difficulty in maintaining engagement with non-coding artifacts. | B05 |
| **Concept-Focused Pedagogical Strategy** | * Students face barriers in conceptual understanding of architecture. * Misconceptions and knowledge gaps persist due to abstract nature of SA. | B06 |
| **Project-based simulation** | * Simulations are resource-intensive to set up. * May oversimplify real-world architectural complexity if not designed well. | B07 |
| **Industry-oriented modelling practice** | * Industry modelling practices vary widely, making it hard to teach a one-size-fits-all method. * Bridging the theory–practice gap is difficult for both students and instructors. | B08 |
| **Scaffolded RE to SA learning** | * Students face challenges with transitioning from requirements to design due to cognitive overload. * Need extensive instructor feedback and iteration. | B09 |
| **Game based Learning** | * Risk of students focusing more on game mechanics than learning outcomes. * Limited transferability of game-based insights to real-world projects. | B10 |
| **Microservices-based problem-solving** | * Requires managing distributed systems complexity in an educational setting. * Students may lack foundational understanding to design microservices effectively. | B11 |
| **Model driven engineering (MDE)** | * MDE tools and formalisms can be overwhelming without strong abstraction skills. * Tool learning curves detract from architecture learning. | B12 |
| **Queueing theory for performance modelling** | * Students often lack the mathematical background to apply queuing models meaningfully. * Perceived as too theoretical or niche. | B13 |
| **Instructional Heuristic in SA** | * Lack of universally accepted best practices for teaching SA. * Adapting heuristics to different teaching contexts remains complex. | B14 |
| **Ranking-based Task Design** | * Students may focus on surface-level comparisons rather than deeper architectural reasoning. * Designing fair and comprehensive ranking tasks is a challenge for educators. | B15 |