**5.1. RQ1: What are the different teaching strategies used in software architecture education?**

Findings and analysis on RQ1 here....

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reviewer initial | Theme | Strategy | Explanation | Reference |
| PG | **Traditional approach** | Teach traditional classes in software architecture education | In this strategy, professors or tutors share their knowledge with students in person. | S29 |
| Lectures/Traditional teaching | Guest Lectures to illustrate the relevancy of topic covered outside the open-source systems as well.  The instructor explained the concept at the beginning of the laboratory session using slides, web pages, and code extracts from 15 to 55 minutes. | S58, S60 |
| TD | Constructive Alignment | This strategy or principle is student-centred approach that combines constructivist learning theories and aligned curriculum to enhance the learning outcomes by encouraging students to adopt deep learning approaches. | S45 |
| TR | Preparing Teaching Manuals | To overcome the multiple teaching methodologies, detailed teaching manuals consisting of learning objectives, outcomes and teaching methodologies for every topic included in the curriculum. | S52 |
| PG | Guest lectures | Brings in guest speakers from the software industry to expose students to current trends, challenges, and expectations. | S67 |
| PG | **Real world-based learning** | Problem-based learning | Students address architectural problems and instructors play minimum roles in this strategy S29, while S42 examines the application of problem-based learning to adult learners focusing on challenges and effectiveness of the strategy without any preset goals and expected outcomes. In S43, it is tailored for computing courses and defined as mapping real-world conceptual models directly to high level component abstraction. This is further refined to analyse their understanding using Neural Pathway Based Learning (NPL) strategies. *(It is a pedagogical technique to use a combination of teaching strategies to create deep understanding, knowledge, and critical evaluation by using techniques such as a multiple-choice quiz, animations, games and puzzle based learning, and repeated learning)*. | S29, S42, S43, S49 |
| TR | Architectural Kata Structure | It is a novel, and exciting learning activities with several exercises.    As a learning activity for students to practice the skills needed to complete their assignments where teacher act as a moderator.  The workshop had two phases: Discussion and Design Phase where students explore the kata problem and design the solution. In Peer Review and Voting Phase, they present a vision of their solution and other students evaluate and vote the solution. | S52, S57 |
| TD | Microservices-based problem-solving | Engages students in collaborative problem-solving using microservices architecture to reflect industry practice. | S24 |
|  | Case-based learning | This strategy involves a real-life scenario that allow students to analyse, apply taught concepts and mitigate potentials adjustments S49. In S42, courses were workshop-based with standard structures such as preset goals and expected outcomes. | S29, S42, S49, S70 |
|  | Uses guided inquiry method and provides more structure during small group sessions.  Implementation of case-based learning the form of bullet-cases, mini cases and descriptive cases. | S49, S59 |
|  | Learning by Doing | Used to teach the students how to be able to write, visualise, and turn the 2D shapes into 3D units manually.  Learners perform one or a few architectural activities based on knowledge acquired before or during the actual experience. | S50, S51 |
|  |  | Interdisciplinary learning | Interdisciplinary training and industry collaborations | S63 |
|  |  | Role-play learning | Role play, Aligns with industry expectations by improving students’ ability to understand and apply different architectural styles. | S64 |
| PG | **Blended Learning** | Online learning | In this strategy, two fundamental modalities were used to deliver learning outcomes namely Massive Open Online Courses (MOOCs) and Small Private Online Courses (SPOCs). | S29 |
|  | Synchronous teaching using Zoom, MS Teams, and WhatsApp. Asynchronous methods included recorded lectures and supplementary learning via platforms like YouTube, NPTEL, SWAYAM, EdX, and Coursera. | S55 |
| PG | Intelligent Tutorial System | This strategy provides personalised learning pathways by tracing each student’s progress. | S29 |
| PG | Flipped classroom approach | Teachers or educators offer lectures or show videos to students before the face-to-face session. | S29 |
| TR | Instructor recorded the videos in their own office not more than 6 minutes and uploaded them to a YouTube playlist which was created as an unlisted. | S60, S62 |
| TD | The course was structured as a flipped classroom, were the  class met two times every week. On the first class-session of each  week, they discussed an important quality attribute along with the  common tactics used to satisfy the quality attribute. For the second class session, students were provided with use cases, quality attributes, and constraints of a system they will be responsible for designing and analyzing. | S14 |
| PG | **Personalized learning** | Patterns-based approach | This strategy proposes a patterns-based approach for designing educational technologies by integrating instructional design patterns with software architecture patterns. This approach bridges gap between instructional design and software architecture. It also improved adaptability and scalability of educational technologies. | S31 |
| PG | Student Ownership of Learning (SOL) | This is a strategy that features a model of learning that blends desirable behaviours of student to increase achievements of students and level of learning which is built on their own experiences, knowledge and view of how the world works. | S34 |
|  | Experiential Learning | Used to fabricate self-structured two-meter-high pavilions using 4mm cardboard where they select the rules and different designs.  The trial-and-error processes were involved and documented all. Experiential learning strategy with four stages: Concrete Experience, Reflective Observation, Abstract Conceptualisation and Active Experimentation.  Experiential Risk Learning Model: Combine Kolb’s Experiential Learning Model (Concrete Experience, Reflective Observation, Abstract Conceptualization, Active Experimentation) and Software Risk Management Process (Risk Identification, Analysis, Prioritization, Planning, Resolution, Monitoring) | S50, S54  S72 |
| PG | Adapted Learning Contract Strategy | With this strategy, course contents covered are reinforced by adapting to specific characteristics of each student to increase performance and chances of passing the course. | S38 |
|  | Scaffolded Requirements to SA learning | Students are scaffolded step-by-step from requirements engineering to architectural design in large-scale, distributed system contexts. | S22 |
|  |  | Virtual learning management system | Moodle represents a great contribution to the educational world since it provides an evolving platform for Virtual Learning Management Systems (VLMS) that became a standard de facto for most of the educational institutions around the world. | S5 |
| PG | **Project-based learning approach** | Project-based learning  Mobile Project-based Learning | This strategy addresses challenges of teaching embedded software to students who have limited electronics background integrating open-source tools and hands-on projects S35, while S36 studied the student’s experience with the group-based project exams. With mobile project-based learning, it creates an interactive and collaborative learning environment S39. | S35, S36, S39 |
| PG | Product-Based Learning (PBL) | Product-based learning strategy for programming assignments requires students to complete assignments individually, grade by a human, employ graphical user interface, have open-ended specifications and resembles a real-world product. | S30 |
| UL | Project-Based Learning (PBL) | Students work on open-source or real-world projects, encouraging them to analyse, design, and evaluate architectural solutions. | S2 |
| TD | 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. | S17 |
| TD | End to End Project-based | Integrates business, architecture, and process instruction using real-world frameworks, flipped delivery, and active learning tools (games, projects), encouraging strategic and holistic software design thinking. | S16 |
| TD | Project-Based Learning through Course Evolution | This strategy reflects an evolving instructional design where students engage in multi-phase, real-client-inspired architecture projects. The course shifted from lecture-heavy formats to hands-on, quality attribute-driven assignments, incorporating tools, frameworks, and active design improvement tasks. | S15 |
| TR | Project-based Learning | Mimic the software development in industry working in groups promoting teamwork.    Students learn to put practice of industrial context complexity through medium complexity projects in a team.    Involves complex challenges and problem-solving skills with real-world case studies which promotes practical skills. | S51, S52, S56 |
| PG | Open-Source Projects | Uses open-source systems as educational material. Students analyze, improve, or document large-scale architectures, gaining insights into real industry-level systems. | S65 |
|  |  | Project-based Learning | Architecture-Centric Pedagogy + Research-Driven + Collaborative Learning and Peer Feedback  Emphasizes skills relevant to mobile/embedded systems development and architectural thinking, aligning with real-world software engineering demands​ | S75 |
|  |  | Platform-based learning | The tool focuses on MVC, one of the most widespread patterns in industry, and aligns with industry expectations for agile development and maintainable architecture. The tool helps bridge the gap between design and implementation—a common real-world challenge. | S83 |
| PG | **Agile based learning** | Scrum | As scrum is a popular empirical process control model designed to manage complex activities, it was found promising to be used as one of the teaching strategies to facilitate learning process in software architecture. | S41 |
| UL | Agile based learning | Students work in iterative cycles (sprints) to develop solutions, reflect on progress, and adapt based on feedback mirroring real-world Agile practices. | S7, S9 |
| TR | Agile Methodologies (Scrum) | Scrum with three phases were implemented. Initial phase: general objectives of the assignments were outlined. Development phase: Sprints (3 sprints) producing one product based on assignment after series of cycles. Closure phase: When all requirements are met, the final product is ready for review. | S48 |
| UL | **Game-based learning** | Gamification | Students engage in gamified activities to practice software engineering tasks, enhancing motivation and understanding through challenges, rewards, and interactive scenarios. | S11 |
| TD | Game based Learning | Uses educational games to simulate architecture design decisions in a motivating and controlled environment. | S23, S20 |
|  | Game-based Learning (Kahoot! DecidArch) | The games support the learning process of specific topics, and it involved the card style, requiring face-to-face interaction.    The Kahoot! was used as interactive test to check the students’ understanding of the concepts from videos at the beginning of laboratory session during flipped classroom methodology.    The DecidArch as used for teaching a particular topic where students play the game in a team of architects and bringing various skills and competencies together. They will play the game at midpoint where the learners attend lecturers and working on their practical project parallel. | S51, S60, S61 |
| P | Gamified architecture education | Involves the use of educational games or simulations (e.g., role-playing games or card games) to teach architectural principles and trade-offs in engaging ways. | S77, S69 |
| TD | Game based learning | Developed a game to more realistically simulate, teach, and stimulate discussion about the process of software design. The game based learning then complement traditional lectures and exercises in software architecture courses. | S20, S23 |
|  |  | Game-based learning | DecidArch v2: a card-based educational game designed to teach software architecture  The game simulates industry-relevant decision-making processes in software architecture, focusing on collaboration, trade-offs, and stakeholder concerns. | S78 |
| PG | **Software system learning** | Usability-Supporting Architecture Patterns (USAPs) | By adopting Usability-Supporting Architecture Patterns (USAPs approach, it helps to tackle the problem of usability flaws in software systems focusing on architecturally sensitive usability concerns early in the design process S37. | S37 |
| PG | Software Product Line Architecture | This is a methodology in which common features are shared with a family of products, addressing business, architecture, processes and organisational aspects. It has proven to be a methodology for the development of software products at lower costs, shorter time, and with higher quality. This is a proposed architecture to develop mobile-learning (m-learning) applications. | S44 |
| PG | Virtual and augmented reality in architecture education | This approach presents an innovative approach by integrating virtual and augmented reality into software architecture education, aiming to provide a more interactive and engaging learning experience. With large-scale systems and advanced visualisation techniques, the paper seeks to present industry practices and tools. | S47 |
| P | Visualization and AR/VR tools | Uses immersive tools such as 3D or AR-based platforms to visualize and interact with complex software architectures, enhancing engagement and understanding | S82 |
| P | Archinotes | Archinotes allows teachers and their assistants to monitor each student’s progress, as well as problems they may have on their project. This information is immediately available to the teacher and allows him to adjust the content of some of his classes to solve specific and recurring problems. Additionally, Archinotes allows students in a team to coordinate their activities and share information using methods that don’t necessarily use face-to-face strategies. | S68 |
| TD | **Abstraction and modelling** | Model-Driven Teaching | Aligns modelling instruction with real-world business needs and professional practices in architectural modelling. | S21 |
| TD | Model driven engineering (MDE) | Emphasises the use of MDE for teaching abstract system representation and transformation logic. | S25 |
| TD | Queueing modelling Approach | Applies performance modelling via queueing theory to teach architectural evaluation techniques. | S26 |
| PG | Abstract modelling | The strategy focuses on abstraction modelling, design patterns and functional decomposition in software engineering which enhances student’s understanding of system design and prepares them for real-world software development opportunities S32. | S32 |
| PG | Abstraction and System Thinking | For this strategy, the study highlights the importance of teaching abstraction and systems thinking to improve proficiency in modelling software systems. | S33 |
| P | Early design-centric education | Teaches architectural thinking early in the programming curriculum using structured modelling tools and methodologies. | S80 |
| PG | Simulation-based Teaching Strategy | This strategy uses many system-engineering concepts of modelling, simulation, visualization, and analysis with the goal to disciplined design, implementation, testing, and evaluation practices in students. | S46 |
| TR | **Collaborative learning** | Team-based Learning | Focuses on teamwork and peer-evaluation through group activities. | S56 |
| PG | Collaborative Decision-making | This strategy explores behaviour of software  engineering students as a beginner software architects in various roles and finds out how they make, challenge and capture a set of an architecture design decision and architectures in a collaborative way. The roles assumed by students is expected to enable and promote critical design thinking for decisions in a better quality and architectures. | S35 |
| TG | Collaborative Learning | Students contribute to achieving a particular goal.    Cross-collaboration through three models: Collaborative Studio Model, Trans disciplinary Encounter Model, Real team Collaboration where students work on their own and then working in team.  Collaborative Learning through open-source projects such as Git and GitHub in groups since the members have different levels of experiences. | S51, S53, S58 |
| P | Real clients or industry experts | Involves collaboration with actual clients or professionals. Provides exposure to realistic constraints and expectations. | S66 |
| p | Conferences featuring industry guests | Brings in guest speakers from the software industry to expose students to current trends, challenges, and expectations. | S67 |
| UL | Computer supported collaborative learning (CSCL) | Students engage in CSCL environments where they collaborate online to complete shared tasks and solve problems. | S6 |
| PG | Apprenticeship learning | This is an educational model in which an apprentice learns a skill or trade through observation and practical experience under guidance of an expert. | S30 |
|  | Collaborative decision-making teaching strategy | The study employs a collaborative decision-making teaching strategy where students assume different roles—senior architects, cognitive architects, and junior architects—to simulate real-world decision-making processes in software architecture. | S40 |

Screened out articles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UL |  | N/A | ( Doesn’t specifiy about teaching strategy) | S3, S10, S12, S13, S19, S27, S18, S73, S79, S5 |
|  |  |  |  |  |
| UL |  |  | Focuses on software engineering course and is not specific to software architecture | S1,S4, S8, S28, S67, S81, S74, S76 S71 |