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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Studies Count in each Category** | **Reviewer initial** | **Theme** | **Explanation** | **Reference** |
| **1** | **UL** | **Project-based Practical Experiences** | Study describes a semester-long Agile project where students develop software, define architectural choices, and iteratively refine design. | **S7** |
| **2** | **TD** | Project-based learning with real-world scenarios involving microservice development. | **S24** |
| **3** | **UL** | The study integrates hands-on learning through a capstone project, simulating real-world Scrum practices. | **S9** |
| **4** | **TD** | Employs a flipped classroom model with iterative SA design assignments and real-world case studies in multiple domains. Employs a flipped classroom model with iterative SA design assignments and real-world case studies in multiple domains. | **S14** |
| **5** | **TD** | Students engaged in practical pre-exam tasks in business process modelling, software functions modelling, and data modelling. | **S21** |
| **6** | **TD** | Involves project-based learning through simulated large-scale software development scenarios. | **S22** |
| **7** | **TD** | Interactive gameplay allowing practical application of architectural decision-making. | **S23** |
| **8** | **TD** | 80% of surveyed courses include hands-on labs or projects. Use of model transformations, code generation, and architectural modelling tools. Some courses include group-based design exercises, especially in capstone projects. | **S25** |
| **9** | **TD** | Provides interactive simulations (e.g., Lego-based Agile exercises), case studies derived from industry projects, and practical application through project-based assignments. | **S16** |
| **10** | **PG** | Provides practical examples through case studies such as adult literacy programs. | **S31** |
| **11** | **PG** | Students engaged in constructing UML diagrams (activity, class, and sequence diagrams), providing practical experience in software modelling. | **S33** |
| **12** | **PG** | Provides hands-on practical experiences through interactive lab sessions and projects. | **S34** |
| **13** | **PG** | Provides hands-on practical experiences through gamified programming exercises and interactive projects. | **S38** |
| **14** | **PG** | The use of Scrum in the course provides students with practical experience in agile methodologies, fostering collaboration and iterative development. | **S41** |
| **15** | **PG** | It mentions that problem drives learning. Therefore, practical skills are based on different types of problems and problem scenarios where students are provided technology and subject content to choose from a number of solutions. Learners are challenged through visual instruction and exercises. | **S43** |
| **16** | **PG** | Students were strongly encouraged to attend both the lectures and laboratory classes during the 12-week course period. This gave them hands-on practical experiences in programming concepts and process of programming | **S45** |
| **17** | **PG** | Offers students the opportunity to interact with and explore software architectures in a 3D environment, facilitating practical understanding through simulation and visualisation. | **S47** |
| **18** | **TR** | Through the architectural design studio and project-based group work architectural design processes, experiencing practical collaboration. | **S53** |
| **19** | **TR** | Involved in realistic hands-on activities, simulating architectural risks related to availability, security, and performance. Design for software qualities during session 2. | **S54** |
| **20** | **TR** | Students work on a project by applying ATAM (Architecture Tradeoff Analysis Method) to evaluate a real-world case. They are grouped by individually selecting a role, and activities are performed in groups, discussing their proposals through peer-review and brainstorming. | **S56** |
| **21** | **TR** | Students apply theoretical concepts to real, by selecting open-source projects hosted on GitHub. Contributes to open-source and collaboratively documenting the works where they meet real software architects and learn from them. | **S58** |
| **22** | **TR** | The flipped-classroom approach allows more in-class time for hands-on practical exercises by delivering lectures via pre-recorded videos. | **S60** |
| **23** | **P** | The experiential component includes hands-on practical activities such as simulating risks and designing countermeasures. | **S72** |
| **24** | **P** | Strongly relevant: students engaged in distributed architecture design tasks simulating industry. | **S76** |
| **25** | **P** | Students gained hands-on experience via template-based applications, working on projects using MVC, TDD, and multi-tier architectures. | **S79** |
| **26** | **P** | The course includes project-based learning, team-based software development, and SDLC practices. Students implement web services and intelligent algorithms. | **S81** |
| **1** | **UL** | **Real-world based experiences** | Students developed a real-world software project (Theatre Management System), using GitHub and incremental development cycles. | **S1** |
| **2** | **UL** | Students are provided hands-on experiences through real-world software development projects, semester-long Agile-based capstones, and simulations of architectural decision-making | **S5** |
| **3** | **UL** | The use of CLPL in master’s courses where students develop real-world software applications | **S6** |
| **4** | **UL** | Students are exposed to software architecture concepts through curriculum-aligned tasks such as building architectures, modelling requirements, designing and testing solutions, and creating maintenance plans | **S10** |
| **5** | **UL** | Students conducted real elicitation with actors, built prototypes, used CI/CD pipelines, and managed real or simulated client interactions. | **S12** |
| **6** | **TD** | It offers hands-on projects in which students tackle real architectural design problems. | **S15** |
| **7** | **TD** | Both project types required students to design, implement, and evaluate software architectures.  Game project students demonstrated more complex architectural designs, possibly due to higher motivation and engagement. Incremental development methodology was used, reinforcing real-world architecture evolution principles. Students faced technical challenges integrating COTS, providing valuable practical experience. | **S20** |
| **8** | **PG** | Training software architects: apply knowledge and skills demanded in the industry and develop essential skills, use job advertisements to align curricula with industry needs, focus on using industry-applied tools, use papers and materials from current state-of-the-art software architecture. | **S29** |
| **9** | **PG** | Provides hands-on experience through projects resembling real-world applications. | **S30** |
| **10** | **PG** | Provides hands-on experience through the development of mobile robots using commercial sensors and actuators. | **S35** |
| **11** | **PG** | Participants were given a task to fix a usability problem in a software architecture design meeting the requirements of the usability scenario. | **S37** |
| **12** | **PG** | The course design incorporates practical exercises and real-world case studies to enhance learners' design skills. | **S42** |
| **13** | **PG** | Simulation architecture allows students to design, construct, and manipulate sensors, actuators, and other components, offering hands-on experience in system integration and software development. | **S46** |
| **14** | **TR** | The course employs hands-on projects, GitHub repositories for version control, Trello for task tracking, and a final project requiring students to develop a blockchain-based NFT marketplace. Students worked with Docker, Kubernetes, API design, CI/CD pipelines, and cloud deployments. | **S48** |
| **15** | **TR** | Students fabricated four self-structured pavilions using cardboard without digital software. Manual material testing and form transformations were central to their learning process. Hands-on activities included unit transformations, connection types, assembly processes, and structural evaluations. | **S50** |
| **16** | **TR** | Project-based Learning: Students work on real-world projects, sometimes in collaboration with companies. Learn-by-doing: Exercises involving software architecture design and evaluation. Game-based Learning: Used for to support the learning process of specific topics (ATAM games). Collaborative Learning: Collaboration involving different stakeholders to enhance problem-solving skills. | **S51** |
| **17** | **TR** | Project-Based Learning (PBL): Students worked on medium-complexity projects, simulating real organizational environments. Architecture Katas Workshop: Students apply their skills in a simulated environment, evaluated by industry professionals. Entrepreneurship Activities: Deliver more realistic products. | **S52** |
| **18** | **TR** | Students work on industry related projects and apply game-based decision-making learning approach where they got to experience the process of decision making and working in team. | **S61** |
| **19** | **P** | Students engage with practical software design and tree-based access control systems. | **S67** |
| **20** | **P** | Students use Archinotes to collaboratively design architectures and construct Software Architecture Documents (SADs), simulating real-world team scenarios. | **S68** |
| **21** | **P** | CSDCT provides an interactive, hands-on experience in designing and correcting client-server architecture models. | **S70** |
| **22** | **P** | Students apply architecture through design (UML), MVC, system structure in real-world app projects. | **S74** |
| **23** | **P** | Students build and extend Prism-MW middleware and implement real-world projects on PDAs | **S75** |
| **24** | **P** | The RPG provides students with an immersive, scenario-based, role-playing environment. | **S77** |
| **25** | **UL** | Advocates for capstone projects, industrial internships, and lab-based learning that simulate real-world tasks. | **S13** |
| **1** | **UL** | **Tool and Platform-based Skills** | Coding challenges, software development projects, version control competitions, and risk scenario analysis.  Use of tools like GitHub, HALO, and custom platforms like Team Feed and Class game for interactive learning | **S11** |
| **2** | **TD** | Students participated in iterative practical tasks involving Client-Server, Master-Worker, Broker, MVC, and Microservices patterns using Google App Engine and IBM Cloud Lite, gaining practical experience with distributed architectures. | **S17** |
| **3** | **PG** | Provides hands-on practical experiences through customizable m-learning applications. | **S44** |
| **4** | **TR** | Introduced software tools and created architectural diagrams and later made optional due to the steep learning curve. | **S49** |
| **5** | **P** | Students engage in hands-on tasks through modelling, pseudocode generation, and coding, using tools like R&G Canvas and visual editors like draw.io. | **S80** |
| **6** | **P** | VisAr3D enables exploratory learning through interactive 3D visualization, encouraging active engagement and manipulation of architectural models. | **S82** |
| **7** | **P** | Students use the tool to model class diagrams, generate Laravel code, and iteratively modify both code and diagrams. It supports active, project-based learning. | **S83** |
| **1** | **UL** | **Collaboration** | Students worked in start-up-like teams, collaborated with real customers, and built software prototypes using contemporary tools and processes. | **S2** |
| **2** | **UL** | Students were immersed in hands-on experiences using GitHub (version control), SonarQube (quality analysis), and Microsoft Project (planning), applying them in team-based development projects. | **S8** |
| **3** | **PG** | By using the modeling, teamwork, proper patterns and corresponding technologies, students can get hands on experience in system development while appreciating the theory of abstraction.  Peer learning, practicing teamwork, and developing interpersonal skills amongst students. | **S32** |
| **4** | **PG** | Provides practical experience through project-based learning and group collaboration. | **S36** |
| **5** | **PG** | Students participated in projects using mobile devices to collaborate and access resources, providing practical experience in software architecture design and implementation. | **S39** |
| **6** | **P** | The study emphasized hands-on learning through in-class activities. Students in the FC group participated in interactive discussions, case studies, and group assignments | **S62** |
| **1** | **TD** | **Analytical Skills** | Evaluates student engagement with architecture documentation while performing architecture-related comprehension tasks. | **S18** |
| **2** | **TD** | Students use queueing models to analyse different architectural decisions. | **S26** |
| **3** | **PG** | Students engage in practical tasks using a unique decision-modelling tool to capture and analyse design decisions, providing hands-on experience in software architecture. | **S40** |
| **4** | **TR** | Encouraged peer discussions to understand architectural trade-offs. | **S49** |
| **5** | **TR** | They were provided through analysing of realistic kata problems, designing architectural solutions and presenting the solutions and peer-review feedback. | **S57** |
| **6** | **TR** | Students gain hands-on practical experiences through case discussions, debates, and public hearings, where they apply theoretical knowledge to solve real-world problems. | **S59** |
| **7** | **P** | Analysing Open-Source Systems and Software Architecture Design and Implementation with evaluation of the quality of the project done | **S65** |
| **8** | **P** | DecidArch is an interactive, hands-on card game that immerses students in practical architectural decision-making tasks. | **S78** |