SOFTWARE ARCHITECTURE COURSE

While students in large colleges study mass amounts of theory, we offer up to date, fresh and relevant Software Architecture classes **focused on practical work methods, adapted to industry needs** so you can penetrate the job market with enough confidence and the right experience to do your job right.

Our classes are taught by industry experts, those who work simultaneously as interviewers and recruiters in high-tech companies and know exactly what it takes to succeed. Each student learns **exactly** what they need to know for their future jobs – for this reason, all candidates are screened and evaluated before admission in order to guarantee the highest level of learning and ensure future career opportunities.

What does this mean for you? You gain the best hands-on experience and pay less money - two birds, one stone.

Our knowledge, your future

Private classes

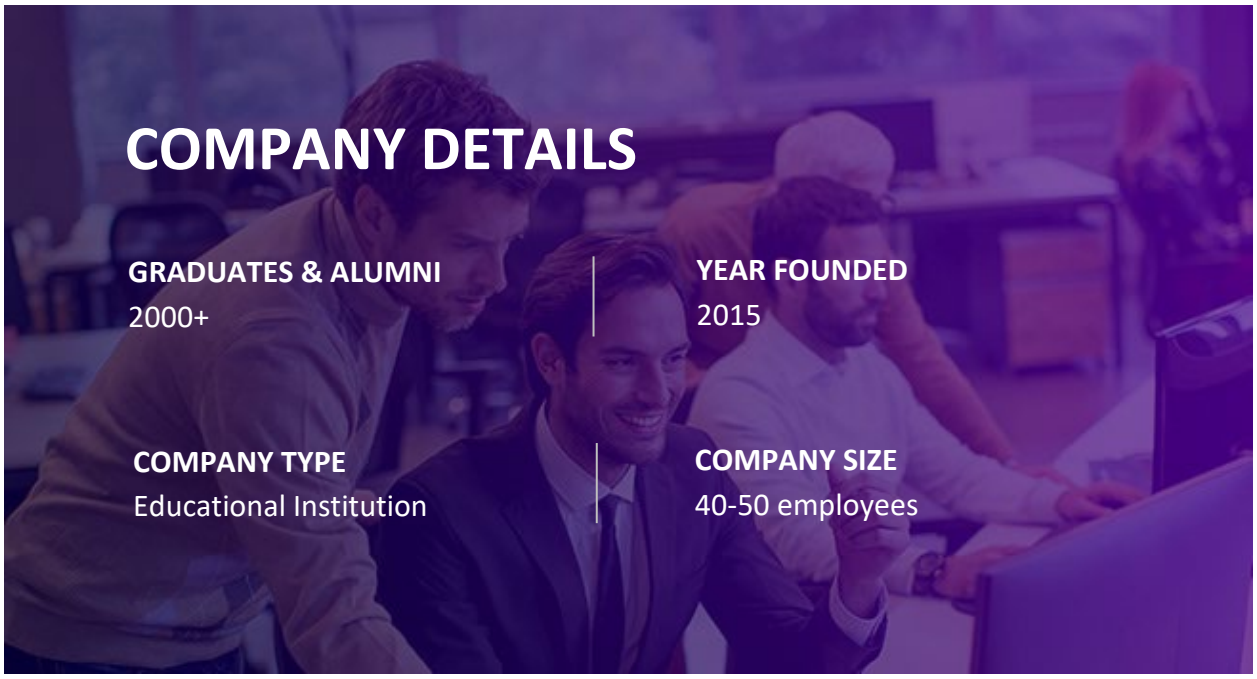
Our Architects courses focus on practical knowledge; in class exercises, homework assignments and learning in small groups which allows for personal attention and better understanding of the material.

Classes for companies

We offer customized Architects courses and workshops according to your company needs. Course materials are suited to your everyday tasks and training requirements.

“Preparation for Work” workshop

We can provide career assistance by reviewing your resume, teaching social media networking and defining LinkedIn content for professional “branding” as well as refer you to relevant positions.



About the course

The Software Architect role requires an extremely valuable but relatively rare set of skills. It combines a deep understanding of software development with a strong ability to prioritize crucial elements, even if it means deprecating redundancies. A good software architect must possess a range of abilities, when the reality is many developers' skills only become more specialized and siloed as they advance.

To stay relevant and successfully expand beyond software engineering into software architecture, you need to think beyond simply writing great code; you must have a macro view of how all the software components work together in a microservices architecture, running on top of containers and platforms managing huge amounts of data in an elegant and efficient way.

In other words, a brilliant software architect must have both the ability to look at and oversee code-writing, but also understand how to manage wider end-to-end processes in a more holistic way.

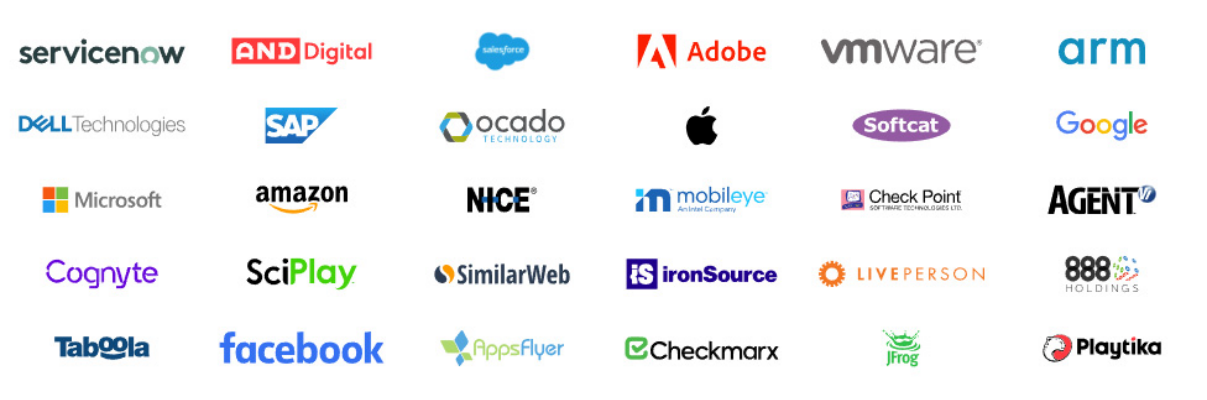
This is exactly what this course is designed to address: it will help you identify and close the gap between being a "one-trick pony" software engineer and being a software architect with a big picture understanding of the best way to structure the software for the company's goals, AND with the practical range to actually bring all of that into being.

Who is this course for?

As we've concluded, qualified Software Architects are in huge demand but not easy to find. Our Software Engineer to Architect course will benefit:

- Backend developers
- Frontend developers
- Team Leads
- Software Architects
- DevOps Engineers
- Data Engineers
- Those with practical coding experience who want to expand their skills and understand how software architecture is done in the leading companies on the market.

OUR ALUMNI WORK WITH THE BEST



Learn from industry experts

Industry-recognized Architects course will teach you current and in-demand skills, ensuring you stay ahead of the curve in a fast-changing industry.

Get hands-on experience

Practical skills are key to succeed and stand out in the market. By working on practical tasks throughout the course, you'll master the skills of a great Software Architect.

Learn amongst professionals

With a network of likewise professionals, enjoy the unique perspective and professional experience of your classmates.

Connect with the industry

Expect dedicated career guidance, access our industry hiring partners, and find your future employment in Software Architecture.

THE INSTRUCTORS

Yuval Wilf
Senior Director of R&D
Cognyte

Danny Gitelman
VP of Architecture
Ritual

David Rotenberg
Senior VP of R&D
Telefire

Aviad Ezra
Principal Software Architect
Microsoft

Arnon Goldstein
Senior Software Architect
Microsoft

COURSE SYLLABUS

Topic	*Description*
Architecture Capabilities	• Architecture decisions • Design principles • Guiding technology decisions • Justifying architecture decisions • Technical skills
Domain-Driven Design	• Using unambiguous language • Defining a Bounded Context • Model Driven Design • Hands-on Modelers • Refactoring towards Deeper insight
Architectural Styles	• Components vs Classes • Component Types • Hybrids & Variants • Scaffolding vs Architecture Patterns • Traditional layered Architecture • Event-driven Architecture • Microkernel architecture • Sacrificial Architecture • Layered architecture
Example Scenario	• Introducing the scenario • Groundhog Day anti-pattern • How to federate a hub • Using dedicated broker instances • Applying a centralized broker • Identify the conditions and constraints • Types of client applications • Analyzing the options • Broker usage and purpose • Message throughput • Internal application coupling • Single point of failure • Performance bottlenecks • Centralized broker approach

Separation of Concerns	• Layers of isolation • Separation of hybrids and variants • Layered architecture considerations • Good general-purpose architecture • Determining a good starting point for most systems • The Architecture sinkhole anti-pattern • Coexistence with Monolithic applications
Architectural Topologies	• Mediator topology • Broker topology • Event-driven architecture • Events and Channels • Impact of Event-driven architecture
Documenting the Process Layer	• Processes exposed as services • Service contracts • Contract creation, maintenance, and versioning • Process availability or unresponsiveness • Reconnection logic on server restart • Addressing failures • The plug-in architecture patterns • Applying business rules and logic
Data Monitoring	• Monitoring Concepts • Introduction to Prometheus stack • Prometheus server • Metrics collection • PromQL • Exporters • Grafana • Alert Manager

Architecture and Continuous Delivery	• Letting the architecture evolve • Discovering iteratively and learn more about the system • Continuous Delivery • Operationalizing abstract architecture • Mature engineering practices • Managing coupling intelligently • The Delivery teams • Constant flow of new features into production • Continuous Integration • Fast, automated feedback • Deployment Pipeline
Evolutionary Architecture	• Enterprise Integration Patterns • Integration Challenges • Coordination challenges • Latency issues • The network and security • Topology changes • Increasing transport costs • The network is not heterogenous
Integration Styles	• File transfer • Shared database • Remote procedure calls • Messaging and invocation • Integration simplicity • System decoupling • System abstraction
Integration Challenges	• Timeliness of data synchronization • Data-only transfer shared database • Dear-universal integration via SQL • Performance bottleneck issues • Schema change and Data ownership issues • Data encapsulation and ownership • Web services • Avoiding tight system coupling

Cloud Architecture	• Amazon Web Services • Regions • Cloud components • Cloud storage • Lambda • Serverless architecture • Runtimes • Templates and blueprints
(Micro) Service Oriented Architecture	• Synchronous and Asynchronous communication • Reliable messaging • Highly decoupled systems • Achieving scalability • Integration beyond the firewall • Cross platform standards • Which is the best service integration style? • Presenting alternatives • Articulate the pros and cons of each
Architecture and Business Strategy	• The Enterprise Operating Model • Business Needs • IT Capabilities • Business Operations and IT Systems • Infrastructure enterprise architecture • Architecture Governance