VLADIMIR BACVANSKI

ARCHITECTURAL THINKING
How to be a good Architect
OCTOBER 10-12, 2016

MODERN INFORMATION ARCHITECTURE
OCTOBER 13-14, 2016

RESIDENZA DI RIPETTA - VIA DI RIPETTA, 231
ROME (ITALY)
The Architectural Thinking course teaches the skills and the mindset required to build robust, scalable and maintainable architectures. The course teaches a methodology and a set of techniques that show how to ensure that all aspects of a technical architecture are covered and documented. We cover a set of principles, criteria, architectural structures and patterns essential in building software and enterprise architectures. The course includes the recent developments that cover scalability, cloud, Microservices, Big Data, and NoSQL impact on software architecture. Numerous discussion and design exercises provide for engaging class.

AUDIENCE

• Developers
• Designers
• Architects
• Consultants
• Technical managers

PREREQUISITES

Experience with software development.

OBJECTIVES

• Upon completion, attendees will be able to:
• Recognize and understand the key qualities of good software architectures
• Be able to recognize, understand and use architectural patterns
• Effectively document architectures
• Be able to identify the various customers of IT architecture and work with architectural views
• Apply an effective methodology for evolving architectures
• Create architectures that withstand the challenges of modern applications
• Understand the impact of cloud, NoSQL, Microservices and container technologies
## 1. Introduction
- What is Software Architecture?
- Architectural categories
- What makes a good architecture?
- Cohesion and coupling
- Software processes and the architecture business cycle
- Where do architectures come from?
- Typical architectural goals
- Enterprise Architecture

## 2. The Architectural Development Process
- A reference process
- Examples of popular processes
- Rational Unified Process (RUP)
- TOGAF Lifecycle
- Agile and Architecture

## 3. Architectural views
- The Stakeholders
- What are views?
- The Zachman Framework
- IEEE 1471-2000
- How to select views
- Commonly used views

## 4. A Notation for Describing Software Architecture
- Why notation?
- UML and beyond
- Other notations
- A pragmatic approach

## 5. Requirements and Architecture
- The requirements view
- Ensuring continuity in architecture
- Documentation of requirements
- Validation of architecture against functional requirements

## 6. Model Driven Engineering
- Model-Driven Engineering
- Productivity through automation
- Domain Specific Languages (DSL)

## 7. Architectural Assets
- What is an Asset?
- Reuse perspective
- Asset management

## 8. Quality Attributes: Modern Nonfunctional Requirements
- What are Quality Attributes?
- Attribute taxonomy
- Examples of Taxonomy
- Faults vs. failures
- Business qualities
- General scenario
- Example, Availability
- Architectural styles and patterns

## 9. Architectural Patterns
- Categories of patterns
- Layers
- Model View Controller Family of Patterns
- Publish subscribe
- Broker
- Blackboard
- Service Oriented Architecture
- Microservices

## 10. New Patterns for Scalability, NoSQL, Big Data, and Cloud
- Patterns for Scalability
- Sharding
- NoSQL Stores
- MapReduce and Hadoop
- The impact of Cloud
- Microservices and Cloud
- Dealing with Failures
- Containers and their Impact: docker and Related Systems
ABOUT THIS SEMINAR

This 2-day course has the objective to teach you the key elements of modern Information Architecture, including Big Data and NoSQL. The course starts with the introduction of the key Information Architecture concepts, and moves quickly to practical data integration and its different approaches and trade-offs. For any Information Architect, data models provide the platform for decision-making and communication. We explore different data models and through them we address various aspects of enterprise data models. We explore best practices and advice on creating data models, including making real-world compromises. Next we explore various options for data and metadata management, including master data models.

The second part of the course will guide you through the new landscape for Information Architecture for NoSQL, Digital, Cloud, and Big Data. You will understand how to architect and manage those solutions, and you will get a good understanding of different pieces that make Big Data solutions, such as Hadoop, and the variety of NoSQL stores. Integration with relational stores is a particularly important part of this picture. Data Lakes and associated practices of governance, security, and data quality are covered as well.

The material is reinforced through numerous discussion questions and design exercises.

AUDIENCE

• Information Architects
• Data Architects
• Data Professionals
• Data Managers with strong technical background

OBJECTIVES

Upon completion, attendees will be able to:

• Make sound decisions about Information Architecture
• Architect Information flows and data integration approaches, including those for mobile applications
• Create different types of data models and understand metadata management
• Make decisions about Big Data and NoSQL solutions
# Outline

1. Information Architecture
   - What is Information Architecture?
   - Information Architecture in context
   - Value of Information Architecture
   - Enterprise Architecture and Information Architecture
   - Information Architecture and Architectural Views
   - Data domains
   - Content Management
   - Predictive Analytics

2. Data Flows and Information Integration Strategies
   - Data flows
   - Data integration strategies
   - Impact of SOA
   - Messaging
   - Services
   - Bulk updates
   - Synchronizing different data stores
   - Integration trade-offs
   - Data integration and mobile applications

3. Data in the Cloud
   - Data solutions and the Cloud
   - Architectures for Cloud data solutions
   - Security risks for data in the Cloud

4. Information Modeling: State of the Art
   - The Value of Modeling
   - Business models/Domain models/Enterprise Data Models
   - Logical data modeling
   - Physical data modeling
   - Modeling for operational stores
   - Modeling for Enterprise Data Warehouse
   - Modeling for NoSQL
   - Modeling for SOA
   - Interplay with Enterprise Architecture models and UML

5. Information Modeling Advice and Best Practice
   - Consistency
   - What to do when models get out of sync?
   - Keeping models in sync: options
   - Precision
   - Mapping between models
   - Normalization options
   - Playing well with Enterprise Architecture
   - Data models and UML
   - Industry specific models

6. Data and Metadata Management
   - Data governance
   - Processes for governance
   - Governance and Agile
   - Data stewardship
   - Metadata Management
   - Master Data Management
   - Security

7. Architectures for Big Data
   - Architectural approach for Big Data
   - Problems with conventional systems
   - Map Reduce algorithm
   - Traditional Database Applications
   - Architectural approach for Hadoop
   - MapReduce
   - Relevance of MapReduce to Big Data
   - Architectures for Big Data
   - Common applications of Big Data
   - Storing and processing unstructured data
   - Data Lakes
   - Implementing Data Lakes
   - Data Lake governance, security, and data quality
   - Big Data technologies for small and medium data

8. Hadoop
   - What is Hadoop?
   - The Hadoop Architecture
   - Hadoop Distributed File System
   - HDFS Architecture
   - HDFS API
   - Scalability
1. Data replication
2. Hadoop Applications
3. Typical Hadoop algorithms
4. Best practices for Hadoop Architecture

9. Big Data Analytics with Hive
   • What is Hive?
   • Hive Architecture
   • Data Warehouse using Hive

10. Big Data Analytics with Pig
    • What is Pig?
    • Analyzing data using Pig
    • Using Pig Latin to build data analysis programs

11. Big Data Analytics with Scalding
    • What is Scalding?
    • Processing Big Data using Scalding
    • Integrating Big Data analytics with Software Development Workflow

12. Architectural Elements of the Hadoop Ecosystem
    • Flume
    • Sqoop
    • Zookeeper
    • YARN: The new MapReduce
    • Stream Processing and Big Data
    • Storm: Stream processing
    • Spark: The next generation Big Data Architecture

13. Hadoop as the new Data Warehouse
    • ETL jobs and Hadoop
    • Integrating Big Data with conventional systems
    • The human factor of Big Data introduction
    • Best Practices of Hadoop/NoSQL introduction

14. NoSQL: Not Only SQL
    • Why NoSQL?
    • Relational database problems
    • Scalability, its Price, and its Limits
    • Key-Value Stores
    • Columnar Stores
    • Document Stores
    • Graph Stores
    • CAP Theorem and its impact
    • Data modeling for NoSQL stores
    • Use cases for NoSQL stores
    • Selecting the right store
    • Polyglot persistence
    • Integrating NoSQL and relational databases
    • Optimal choices and reasonable compromises
    • Integrating with Conventional IT
INFORMATION

PARTICIPATION FEE

Architectural Thinking
€ 1700

Modern Information Architecture
€ 1300

Special price for the delegates who attend both seminars:
€ 2800

The fee includes all seminar documentation, luncheon and coffee breaks.

VENUE

Residenza di Ripetta
Via di Ripetta, 231
Rome (Italy)

SEMINAR TIMETABLE

9.30 am - 1.00 pm
2.00 pm - 5.00 pm

HOW TO REGISTER

You must send the registration form with the receipt of the payment to:
TECHNOLOGY TRANSFER S.r.l.
Piazza Cavour, 3 - 00193 Rome (Italy)
Fax +39-06-6871102
within
September 26, 2016

PAYMENT

Wire transfer to:
Technology Transfer S.r.l.
Banca: Cariparma
Agenzia 1 di Roma
IBAN Code:
IT 03 W 06230 03202 000057031348
BIC/SWIFT: CRPPIT2P546

GENERAL CONDITIONS

DISCOUNT

The participants who will register 30 days before the seminar are entitled to a 5% discount.

If a company registers 5 participants to the same seminar, it will pay only for 4.

Those who benefit of this discount are not entitled to other discounts for the same seminar.

CANCELLATION POLICY

A full refund is given for any cancellation received more than 15 days before the seminar starts. Cancellations less than 15 days prior the event are liable for 50% of the fee. Cancellations less than one week prior to the event date will be liable for the full fee.

CANCELLATION LIABILITY

In the case of cancellation of an event for any reason, Technology Transfer’s liability is limited to the return of the registration fee only.

VLADIMIR BACVANSKI

ARCHITECTURAL THINKING
Rome October 10-12, 2016
Residenza di Ripetta - Via di Ripetta, 231
Registration fee: € 1700

MODERN INFORMATION ARCHITECTURE
Rome October 13-14, 2016
Residenza di Ripetta - Via di Ripetta, 231
Registration fee: € 1300

BOTH SEMINARS
Special price for the delegates who attend both seminars: € 2800

If anyone registered is unable to attend, or in case of cancellation of the seminar, the general conditions mentioned before are applicable.

first name ...............................................................
surname ...............................................................
job title ............................................................... organisation ............................................................... address ............................................................... postcode .............................................................
city ........................................................................ country ............................................................... telephone .............................................................
fax ........................................................................ e-mail ...............................................................
Vladimir Bacvanski, is founder of SciSpike. He is a consultant and mentor on advanced software and data technologies and specialises in NoSQL and Big Data technologies. He has over twenty years of experience with software technologies in areas such as architecture and design of mission critical and distributed enterprise systems, rule-based systems and languages, modeling tools, real-time systems, agent systems, and database technologies. Mr. Bacvanski also has extensive experience in software architecture and requirements analysis, and has helped many companies to select, transition, and apply new software technologies. He is published worldwide and is a frequent speaker, session chair, and workshop organizer at leading industry events. He is also the recipient of a number of prestigious academic scholarships and grants. Mr. Bacvanski received a Doctoral degree in Computer Science from Aachen University of Technology (RWTH Aachen) in Germany.