Complex Challenges?

Some companies always seem that one step ahead: they adapt faster, their products last longer, their performance keeps improving, they stay ahead. In short: they run.

Their edge? Well-architected Systems that are safe, reliable, robust, scalable, and easy to maintain, helping to accelerate innovation and boost competitive advantage!

Are you planning to put Architecting at the heart of your engineering?

Then check out **ADEPT** - Architecture Driven Engineering Processes and Tools, *the* practitioner's methodology designed to help you repeatedly produce...

... elegant Solutions!

Marco Di Maio



ADEPT

marco.dimaio@jckt.com

© 2026 Marco Di Maio

ABOUT THIS COURSE

This three-day intermediate level course covers the WHY, WHAT and HOW of Systems Architecting:

- Understanding the WHY: The need for employing architectures to help cope with the overwhelming nature of complexity.
- Explaining the WHAT: Architectures cater for different system perspectives. They are ideal vehicles to structure understanding.
- Introducing one HOW: Modelling System Architectures using the ADEPT Methodology and it's COFLP Framework.

How does it differ from other Systems Architecting courses?

- ADEPT has been devised as an easy-to-learn and tool-agnostic methodology to cater for any kind of System
- The learning is very intense: over 75% of the time is spent working in teams – either in the Workshops or solving Problems.
- The teams work on the same, real-life challenge throughout the course and have to actually create all Architectures.
- The artefacts created can serve as entry points for transferring the learning to their own work.
- As a back-up, master solutions for each module are provided.
- Part of the practical exercises are deep discussions among the attendees, closely watched and moderated by the trainer.
- Marco is an experienced trainer with almost 25 years of experience, some of which as professor of Systems Engineering.

© 2026 Marco Di Maio

PREREQUISITES & OUTCOMES

Prerequisites

- What to bring to the Course?
  - Being interested, open-minded and curious
  - Sound experience with System Development
  - A basic understanding of Systems Engineering
  - A desire to maximise the efficacy of Development
  - Some experience in working with Requirements

- Who should attend?
  - Engineers with a rudimentary SE understanding
  - Engineers with experience in development
  - Engineers working on a System
    (not necessarily on the “top-level”)
  - Lead and Senior Engineers interested in SE
  - Systems Analysts and Product Owners
  - Requirements Engineers looking for answers
  - Engineers with the job title “Systems Architect”

Learning Objectives

- What is Systems Architecture and how does it differ from the Systems design?
- How does it relate to Systems and Holistic Thinking?
- What makes a good Architecture : how many architectures do we need?
- What makes good Architecture so valuable in the context of complex challenges?
- Why are Interfaces such a fundamental concept in Systems Architectures?
- Why is it important to develop Variants and control Variation Points?
- How do Variation Points relate to Interfaces?
- How does Architecting help with decision making in the context of uncertainty?
- How do the individual Architecture Elements map to one another?
- Why is element-mapping a powerful mechanism to control complexity?
- What is MBSE? How can you model a tool-agnostic architecture?
- How can we ensure that the models are consistent and connected?
- What options do you have to customise Systems Architectures?
- Why is the Logical Architecture so valuable? What is the value of the Physical?
- What are the main skills of a senior Systems Architect?

© 2026 Marco Di Maio

AGENDA

"Time is an illusion - lunchtime doubly so" (the Hitchhiker's Guide to the Galaxy)

Time	DAY 1	DAY 2	DAY 3
08:45 - 09:00	**Trainer is available for discussions**		
09:00 - 10:00	Introduction: Systems Thinking, Architectures and Architects	Functional Architecture	Logical Architecture and Logical Interfaces
10:00 - 10:15	**Break**		
10:15 - 12:15	Context Analysis	Functional Architecture	Logical Architecture and Logical Interfaces
12:15 - 13:15	**Lunch**		
13:15 - 14:15	Operational Architecture	Deriving Operational, Functional, and Performance Requirements	Physical Architecture, Integration and Physical Interfaces
14:15 - 14:30	**Break**		
14:30 - 16:30	Operational Architecture	MBSE, Architecture Frameworks and Deriving Requirements	Physical Architecture, Integration and Physical Interfaces
16:30 - 17:15	**Trainer is available for discussions**		Wrap-up, Feedback and Discussion

© 2026 Marco Di Maio

DAY 1

Systems Thinking and Architecting

- Systems and Complexity
  - Why do we talk about Systems? What are they and what distinguishes them from a 'mere' product?
  - What do people value in a System and how can we develop for it?
  - Why is Complexity the key to understanding System Development?
  - How can we understand and thus – hopefully - control Complexity?
- Systems Thinking
  - Why do we need to think differently?
  - Why are (theoretical) viewpoints so powerful in analysing Problems?
  - How can we then perform a deeper analysis of the challenges?
  - Why is understanding so fundamental to generate value in SE?
  - How do perspective link to artefacts from the development process?
- Architecting
  - What are the various aspects of the Challenges and or the System that define the development approach and help to take the right decisions?
  - How do Architectures relate to these aspects or perspectives?
  - What Architectural Frameworks exist and what are they addressing?
  - How can Architectures improve the understanding of, relationships within, and the decision making for, our development tasks?
  - Why are models to visualise Architectures so powerful? Who benefits and why?

ADEPT Methodology and the COFLP Framework

- The ADEPT Methodology comprises
  - a logical Architecting Process to help create good Architectures
  - the COFLP Framework strictly separating concerns while providing semantically justified relationships between the 5 Architectures
  - a Modelling Tool using a semantic graph engine for max flexibility.
- COFLP Framework
  - Comprises 5 Architectures: Context, Operational, Functional, Logical and Physical that are orthogonal to one another. They represent the essential selection of Perspectives discussed in Systems Thinking
  - COFLP strongly separates concerns by rigorously separating:
    - Context and System-of-Interest
    - Requirements from Constraints: Though constraints are a form of Requirements in the literature, they serve a very different purpose from e.g. Functional or Interface Requirements
    - The IDEAL from the REAL System
    - Operations from Functions: the Operational Architecture into the White-Box view of the System to explicitly allow to take control logic decisions early on; Functions are not seen their extension, or lower levels of Operations but developed as very important System elements in their own right.
    - Logical from Physical Architecture: though both are developed in some form of synchronisation, they are decoupled vigorously to minimise the feedback loops in development.

© 2026 Marco Di Maio

DAY 1

Contextual Architecture (C)

- General
  - Black- and White-Box are standard perspectives on a system. They allow to identify and thus define, what is Context (and therefore out-of-scope) and what has to be developed (in-scope), respectively.
  - The Contextual Architecture is the basis for developing the Customer Requirements. The Items are always kept in their original form and can take any format (textual, graphical, numeric) to allow appropriate Stakeholder Communications
- Structuring the Context
  - Why does a structured approach to Context Analysis significantly help to understand the intricacies of complex challenges?
  - What is the value of Black- vs White-Box Analysis? What do the two Perspectives have in common? How do they change with a higher-level view and how over the Lifecycle of the System-To-Be?
  - How do the various Stakeholders interact with the System-To-Be?
- Modelling
  - Why is a Contextual Architecture an important part of the overall System Model? How does it help to perform a structured elicitation of the Customer Requirements; and how will can we record the Items to preserve their original context (incl. their relationship to the System-To-Be and their Source) and allow for their lifecycle management?
  - What is the role of the Contextual Architecture in the quest for full traceability of lower-level Elements to the Customer Requirements?

Operational Architecture (O)

- General
  - Is the principal Architecture for explicit Decision Making , comparing Data, and interpreting Information, and thus represents a blueprint for the System Control
  - Helps identifying potential for parallelization of tasks to react faster to stimuli and allows discussing potential failure to identify Additional (Alternative, Clean-up or Follow-Up) Operations that might be required
- Modelling
  - How does the Operational Architecture relate to the Contextual (and later in the course: other) Architectures? How can the Items developed for it be connected to those from other Architectures?
  - Which Items do we need to model the Operational Architecture and what is an elegant and easy-to-learn way of visualising them?
  - Why would we separate the temporal, situation and event based Perspective from the Functional? How does it help in controlling the execution of Functions and the communication with the “real world”?
  - Why are Variations Points so important?
  - What is the relationship between the operational Architecture and parallel developments, e.g. of Digital Twins for the System?

© 2026 Marco Di Maio

DAY 2

Functional Architecture (F)

- General
  - The Functional Architecture captures the Analysis of the required qualities and relationships of Object Flows through and within the System.
  - It also defines the Functions that are converting them or – more generally – change the State of Objects in general.
- Modelling
  - How does the Functional Architecture help to monitor the evolution of Objects through the System?
  - What are the rules for modelling the Functional Architecture?
  - What is the fundamental difference between “Signal Flows” and “Object Flows”?
  - Why are Variations Points so important?
  - How does the Functional Architecture relate to the Operational and how can their development still be decoupled?
  - What is the value of a Functional Breakdown Structure? What are the dangers associated with it?

Flex-Module: MBSE, Frameworks & Deriving REQs

- MBSE
  - Why modelling?
  - What is MBSE? What should it be? Where are we now?
  - Why is it still not a reality in product development?
- Architectural Frameworks
  - Conventions, principles and practices
  - Examples from history and present.
  - ARCADIA & Capella – a different approach
- The Decisions captured in the ARCs is used to write REQs
  - The analysis performed during the architecting process is the basis for writing System Requirements
  - Customer, Operational, and Functional REQs can be derived directly from their respective Architectures (i.e. the elements within them)
  - The complete documentation can be (semi-) automatically generated always reflecting the latest versions of the REQs
  - Helps to reduce errors, identify risks and reduce costs.

© 2026 Marco Di Maio

DAY 3

Logical Architecture (L)

- General
  - The System has to be made more manageable by “breaking” it down into Sub-Systems and – ultimately - components.
  - The development of the Sub-Systems has to be decoupled to allow for an effective collaboration of the teams.
  - Interfaces are vital System Items; they are often at the heart of failure.
- Breaking down the System
  - Do we actually need to break down the System?
  - Are we really “breaking” down the System? Is it at all possible?
  - What would be good – logical – aspects to define the breakdown?
  - How can we make sure that the Sub-Systems are still being developed in the best interest of the overall System performance?
  - How can we define the new Items? How can they be modelled?
  - When do we stop to “break it down”?
  - How to write Component Specs?
- Interfaces
  - Understanding IF is vital for success (Projects fail at the interfaces)
  - How can we optimise the number of Interfaces? What tools are available to help with the decisions?
  - How are they managed over the System lifecycle? Allow trade-off analysis for the Architecture Variants

Physical Architecture (P)

- General
  - Understanding the importance and value of Constraints
  - Illustrating the fundamental difference between IDEAL and REAL System and showing the mechanisms to connect them
  - Sourcing the Parts as (C)OTS, by re-use, or by developing new from scratch against the Component Specs.
  - How can Physical Modularity help to drive or control the development of the IDEAL architectures?
- Integrating functionality into a given solution space
  - The integration challenge requires a different mindset from developing a working solution. How can we better prepare and support it?
  - Why are Physical Diagrams so valuable for integrating and Verifying
  - What is “Functional Zoning” and when can we start with it?
- Physical Interfaces
  - Physical Interfaces and how they differ in nature from the other Interfaces found in the IDEAL System
  - Can we map the various types of Interfaces, or are they possibly not related at all?
  - What is the role of the IDEAL and the REAL Interfaces in designing modular Systems?

© 2026 Marco Di Maio

FEEDBACK RECEIVED

NOTE: The course has been running in various formats (2 and 3 days, public and in-house) for 5 years now. It has consistently received positive feedback from the attendees. The most recent (Open course, 3 days, 8 attendees, Zürich, Oct26) is given here, unfiltered:

Usefulness of the workshops

- Good not only for understanding (fits people who learn better as a practical approach), but also to create dynamism to kill monotony — actual “brain-wakers”.
- A workshop/demonstration of implementation of an architecture in a tool would be interesting, even though current tooling is poor as discussed frequently during the training.
- Organisation of WS was ideal. (small groups, one question to class, then group work).
- Got a broader view of systems and learned how to communicate (maybe better with different disciplines)
- I might have liked more diversity in the examples, but on the other hand it is more fun to work on a single system and expand.
- Very good way to deepen the understanding of the concepts, interactive. I also appreciated the discussion between teams at the end to compare results/thinking.
- Very good balance between exercises and content.
- Makes concepts tangible and creates routine in using the tools.

Value of the course for yourself

- Due to current position: Probably unable to directly apply at the top level. However, couple concepts can be
- plug-and-play at my competence level.
- On top of course material, anecdotes, tips, book/conference recommendation were welcome and a lot of
- homework for the next months!
- I will recommend this course, very valuable.
- Logical, and operational architecture can be directly applied/transposed to SWE. Physical architecture could
- be adapted to low level drivers for example. I also feel more comfortable communicating architecture
- concept at system level with system architect or other competency groups.
- I learn a lot. Now I just need to implement it!
- All types of architecture except physical were new to me, see many possible applications in my company.

Appreciations

- Very good course, very good delivery.
- Workshops really help to get up to speed!
- Good communication and very capable of keeping audience awake. No laptop approach works really well in
- this high-stimuli society.
- The content size was adequate for the duration of the training, adequate pace. A lot of interesting discussions
- Appreciations
- with Marco and other attendees.
- Interactive approach to learn the stuff helps a lot.
- Thanks a lot for the training! It was more interesting, fun that what I expected!
- Exercises were really good and very interesting because each exercise build on the previous one.
- Interactive, helpful, inspiring, close to reality.

© 2026 Marco Di Maio