Corso Governance

Course Outline

Table of contents

Introduction

Context

In recent years, the IT landscape has been changing considerably. After an initial phase of exploration, digital transformation is becoming the goal of every company and organization that aims to survive in an increasingly growing market. Concepts, technologies and methods are becoming commonplace and their adoption has been the challenge of the last decade. As if this was not enough, the global crisis of 2020 has accelerated more than ever the need to adopt technologies and skills that allow companies to face the digital world and develop new technological and business paradigms, thus trying to satisfy the consumer in increasingly effective, efficient and innovative ways.

Mia-Academy

Mia-Academy was created to address this challenge.
Mia-Academy is a proven professional culture empowerment system, in which Agile methods are taught and used.

Mia-Academy's goal is to make teams autonomous in the development of new generation digital platforms.

The following document illustrates the training path for the Mia-Platform Governance Professional certification, dedicated to those who want to manage a digital platform based on the Mia-Platform Console.

Course goals

The course objective is to train professionals to define a clear governance of APIs and Microservices. Without it, developer teams may find themselves working completely independently from each other, as if they belonged to different companies, each one with its own software design and development style.
APIs reusability is another mandatory rule to follow in order to avoid creating spaghetti APIs architecture, in which different teams could work on the same feature separately. To do so, it must be crystal clear who owns an API, who develops it, evolves it and what is its meaning. The microservice code is as important as the exposed API, in fact, guidance on the structure of the services must be dictated in order to preserve a common style and quality of the code. Finally Governance over team composition and organization is crucial, having autonomous feature teams help avoid dependency problems and boost productivity.

Modules

Module 1: Data Governance

In a data-centric world it is a mandatory rule to have a strict governance of data, where it resides, the shape it has and who can access it.
This first module explores all the risks of poor data governance and focuses on data management, particularly on **Data Schemas, Security** and **Privacy**.
Mia-Platform Console can help you govern your data and store and expose it via API and Streams.

- Introduction to Data Governance
    - Entities, models, schema definition and data governance processes and roles
- Data Schemas
    - Manage data Schemas in JSON and their life cycle
- Data Security
- Data Privacy
- Mia-Platform Console for Data Governance
    - Overview, crud, api and events
- Data Storage on MongoDB
    - Where to store the data schema (centralized or distributed), CRUD, Single Views
- Data Publishing via API
    - Using API adapter on data model
- Data Publishing via Events and Streams
    - Fast Data

***Takeaways**:*

- *Know how to manage data schemas, security and privacy*
- *Know how to use Mia-Platform Console to govern your data*
- *Know how to expose you Data via API and Streams*

Module 2: API Governance

The second module focuses on the exposure of the data we learned to manage in the first module. An API has its own life cycle based on defined steps and best practices. An API must have a scope, must be documented, tested, versioned, secured and monitored, just as a microservices would be. Once the API is deployed and can be contacted it should be exposed with a **Developer Portal**, that is a centralized tool for API documentation and exploration.

- API domain boundaries
    - Business use case definition
    - Domain Boundaries separation
    - Services and API definition
- API as a product
    - Strategy
    - Design
    - Documentation
    - Development
    - Testing
    - Deployment
    - Versioning
    - Security
    - Monitoring
    - Discovery and promotion
- Developer Portal
    - Expose API documentation
    - Expose API catalogue
- API ownership
    - Who is responsible for API creation, maintenance and usage
    - Definition of an API product manager role

***Takeaways**:*

- *API Definition, Development and Maintenance guidelines and best practices*
- *API Security standards*
- *Know how to create an API Product and how to get value from it*

Modulo 3: Cloud Native Application Governance

In the third module we cover Governance of Cloud Native Application, standards and best practices will be listed and explained. It's crucial to have common guidelines in services development to preserve coherence and quality standards over different teams. To do so, **patterns and architectures,** like pub/sub, saga, backend for frontend etc., exist and help the development team to keep working in the same direction. This module will explain how the Mia-Platform Console can help define a governance plan with **Tenant**, **Project Template**, **Microservices Marketplace** etc.
After the services are deployed the job is not done yet, it's important to have **dashboards** and **alarms** to monitor the correct execution of the services, again Project Template helps define a standard set of dashboard for each newly created project.

- Cloud Native Guidelines
    - Cloud native principles
- Microservices Architectural Styles
    - Architectural style, BFF, Topic, SPA, Application, Gateways, pub/sub, saga
- Mia-Platform Console Introduction
    - Using Mia-Platform Console to define microservices architectures
- Configure Project and Govern them
    - Template, tenant, group projects, label etc
- Runtime Environment and Deploy strategies introduction
    - Manage environments and deploy strategies
    - Store secreted and public variables on the console project and pass them in the service deployment
- Monitoring Cloud Native Applications with Mia-Platform Console
    - Using standard Dashboard to monitor services metrics
- Using Mia-Platform Marketplace to enforce coding guidelines

***Takeaway**:*

- *Define Cloud Native Application principles*
- *Define Architecture and Pattern for microservices*
- *Know how the Mia-Platform console can help govern application development*
- *Know how to monitor deployed applications*

Module 4: Team and Organization Governance

The last module will cover Team and Organization governance, different types of team structures (Feature, Component, Service etc.) and different types of Organizational models with real use cases.
The module will provide evidence on how the Mia-Platform Console can help organize team members and their permission with the Console functionalities. Finally, some Mia-Platform public cases will be explored and analyzed in order to extract some patterns and useful metrics.

- Introduction to Organization Models
    - Agile Scaling (Less, SAFe, Scrum of Scrum, Spotify Model), Feature Teams, Service Teams (Ops Team, UX Team , Mobile Team ...), Governance Teams, Pros/Cons of Team organization models
- Team organization models from the trenches
    - Spotify Story, Netflix, Cattolica, Helvetia Story, AWS Story, Mia-Platform Story
- Organize Teams with Mia-Platform
    - Roles, Permission, Project Hierarchy
- Some examples of Team Organization using Mia-Platform
    - Cattolica, Helvetia

Takeaways:

- *Know how to organize teams in order to get the best performance*
- *Know how to structure your organization*