

REACH THE  
NEXT LEVEL



HANDS-ON WORKSHOPS



<b>FOREWORD</b> › Page 4/5	<b>MACHINE LEARNING</b> › Page 6/7	<b>ARCHITECTURE</b> › Page 8/9
<b>WEB DEVELOPMENT WITH REACT AND REDUX</b> › Page 10/11	<b>GIT</b> › Page 12/13	<b>WEB DEVELOPMENT WITH ANGULAR</b> › Page 14/15
<b>QT/QML</b> › Page 16/17	<b>CLEAN CODE</b> › Page 18/19	<b>UNIT TESTS</b> › Page 20/21
<b>CONTINUOUS INTEGRATION &amp; DEPLOYMENT</b> › Page 22/23	<b>BLOCKCHAIN</b> › Page 24/25	<b>USER EXPERIENCE</b> › Page 26/27



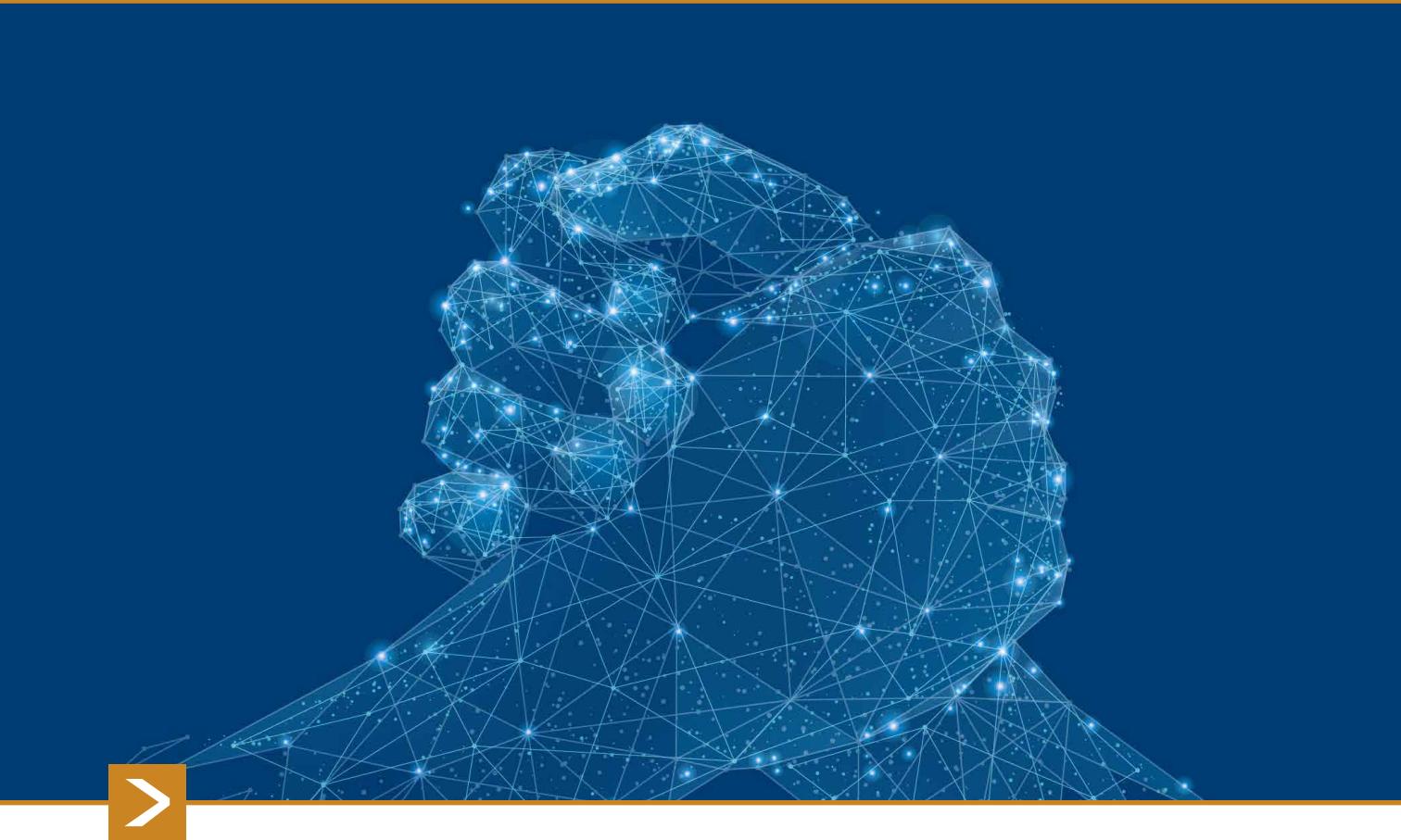
### Coaching Days

Book one or more coaching days after a workshop to deepen your know-how and discuss specific challenges you face in your daily project work with your trainer.



### Laptop required

You will need a laptop for the exercises. The trainer defines the required development environment together with you.



Trust in your abilities and quick access to expertise, lay the foundations for your success in projects and teamwork.

**Knowledge quickly becomes obsolete.** You will need an individual skillset since methods and tools are continually changing, and new technologies pave the way for unexpected opportunities.

Keep your know-how up to date!

**Stay willing to learn.** As practical as possible.

Method Park's workshops support you. You learn modern methods and state-of-the-art technologies - hands-on and **very close to your everyday project work.** Practical exercises provide you with the routine you need in your projects.

The hands-on workshops are **exclusively and individually** tailored to your needs and deliver you and your organization the optimal training solution.

Method Park's hands-on workshops provide the tools you need for the various aspects of software and systems engineering. Qualify your expertise and know-how. Invest in your future.

The Method Park workshop trainers support you – hands-on!

**Target audience**Software developers,  
data scientists**Duration**

2 days

**Prerequisites**Basic programming  
skills**Questions?**workshops@  
methodpark.de

# MACHINE LEARNING



Artificial intelligence and machine learning are established in many industries.

Both techniques were the key success drivers of numerous achievements in autonomous driving, medical image processing, and materials testing.

In the future, Artificial Intelligence (AI) will play an ever-greater role in numerous industries and scenarios. Organizations must acquire at least the basics of AI and machine learning to stay fit for the future.

The machine learning workshop provides these fundamentals. It bundles the flood of information and offers you a compact overview of machine learning theory and practice.

You receive the hands-on knowledge you need to unlock the enormous potential of artificial intelligence for your product portfolio and value chain.

The workshop is highly practice-oriented: half of it consists of applied exercises on a continuous topic. You learn about the appropriate methods and practices for machine learning while focusing on artificial neural networks, the basis for deep learning.

**Basics**

- › Introduction and definition of terms (machine learning, artificial intelligence, big data, etc.)
- › Presentation of technologies used in the workshop, e.g., TensorFlow and Keras

**Landscape of machine learning methods**

- › Supervised learning
- › Unsupervised learning
- › Reinforcement learning

**Single-layer neural networks (perceptrons)**

- › Biological motivation
- › From biological to artificial neurons
- › Learning: optimization, gradient descent
- › Classification of multiple classes

**Basic terms & tools**

- › Loss functions
- › Performance metrics
- › Data partitioning
- › Feature extraction, dimensionality reduction
- › Overfitting and countermeasures

**Multilayer neural networks**

- › Backpropagation
- › Deep learning

**Optional**

- › GPU-accelerated training in the cloud
- › Deploy trained networks on embedded hardware
- › Convolutional neural networks

**Target audience**

Software developers,  
data scientists

**Duration**

2 days

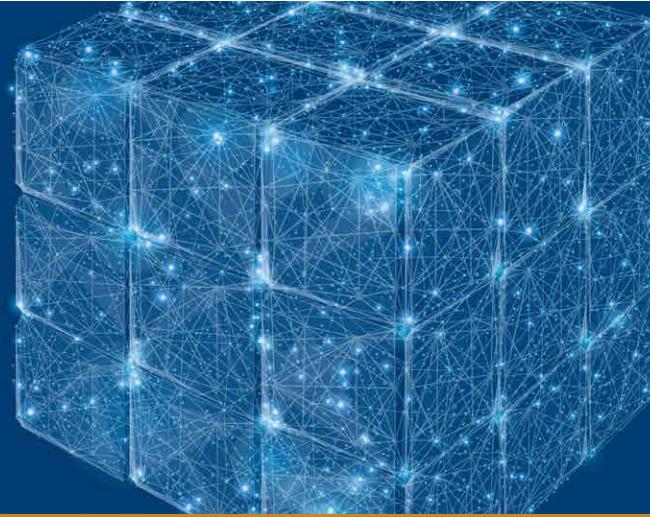
**Prerequisites**

Experience in  
software development

Questions?

workshops@  
methodpark.de

# ARCHITECTURE



How to establish sustainable architecture development in your software development?

Should your architecture support agility, flexibility, and changeability? This hands-on workshop shows the principles and practices for developing and evolving architectures in an agile environment.

The workshop focuses on practice-oriented knowledge transfer. In concrete exercises, you deepen your acquired know-how and apply it with a consistent example: either with a prepared example case or with your current product development integrated into this workshop's practice part.

**Architecture and the architect's role**

- › What is architecture?
- › The role of architecture in agile development
- › How evolutionary architecture enables the continuous development of products

**Communication of architecture**

- › How to communicate architecture to target groups?
- › Agile approaches to documentation
- › Docs as code

**Influencing Factors**

- › Identify and analyze architecture-relevant requirements and framework conditions
- › How to prioritize influencing factors and architectural decisions?

**Design and architecture principles**

- › Design and architecture principles as the key to flexibility and changeability
- › How to decouple the technical core from specific technologies with clean architecture?

**Architecture and continuous delivery**

- › Continuous delivery as the basis for evolutionary architecture

**Target audience**

Software developers

**Duration**

2 days

**Prerequisites**

Knowledge of JavaScript and HTML

**Questions?**workshops@  
methodpark.de

# WEB DEVELOPMENT WITH REACT AND REDUX



How to implement modern and complex web applications with React and Redux fast and easy?

React, Facebook's JavaScript library, makes the creation of interactive user interfaces easy and fast. With React, you create simple view-layers for every application state and instantly implement even complex user interfaces.

Redux is a state container for JavaScript applications that enables centralizing the application states and logic. With Redux, you also unlock high-performance tasks or additional features for your application.

For React and Redux, you have to run unit tests. This hands-on workshop presents you with the appropriate testing tools, e.g., Jest or Enzyme.

The workshop focuses on practice-oriented knowledge transfer, and over half of it consists of hands-on exercises on a consistent topic. You acquire all aspects of React and Redux by working on specific tasks. Your team can use the learned techniques in practice as part of their daily routine.

**React**

- › Which concept is React based on?
- › How to use JSX to define markups in React components?
- › Which methods to use for the implementation of component logic?
- › How to process and define events?
- › What are the best practices?
- › What are additional advanced methods?

**Unit Testing**

- › What are the options for testing implemented React components?

**State Handling and side effects**

- › How to use Redux to manage application states?
- › How to check implemented Redux logic for validity with unit tests?
- › How to integrate Redux with React?
- › How to use the Redux saga to implement side effects such as user interactions or network communication?

**Target audience**

Software developers and integrators

**Duration**

2 days

**Prerequisites**

with or without experience in version control systems

Questions?

workshops@  
methodpark.de

# GIT



Take your software development project to the next level with Git!

Do you want to benefit from distributed version management to assure quality in a regulated environment together with code reviews?

This hands-on workshop accompanies you during the introduction of Git. The trainer teaches you how to use Git and code review solutions, such as Gerrit, GitHub Enterprise, or TFS, and supports you in selecting the right tool.

The workshop focuses on practical knowledge transfer. Together with you, the trainer develops a workflow tailored to your individual needs, covering all phases of development: from check-in over code reviews linked to a continuous integration system to release and life cycle management.

**Git**

- › Basic version management
- › Understand software histories
- › How to create development lines (branches) and versions (tags)?
- › Integration strategies (merge vs. rebase) to merge diverging development lines
- › Synchronization with other repositories (push, pull)
- › Best practices for more efficient development processes with Git

**Code Review with Gerrit**

- › Goals and basics of code review
- › Deliver code for review
- › How to incorporate code review comments?
- › How to solve conflicts?
- › Best practices
- › Collaboration workflows (fork & branch, pull requests)
- › Resynchronization with progressive development
- › How to structure teams and organizations?

**Tools**

- › Gerrit
- › GitLab
- › GitHub (Enterprise)
- › Azure DevOps / TFS
- › Bitbucket

**Target audience**

Software developers

**Duration**

2 days

**Prerequisites**Experience in JavaScript  
and HTML**Questions?**workshops@  
methodpark.de

# WEB DEVELOPMENT WITH ANGULAR



Angular is the top performer among frontend frameworks and established as a standard in the enterprise environment.

Angular is still an indispensable part of modern web development even though some alternatives are available now. With the release of Angular2 and the separation of AngularJS, the framework has undergone a fundamental transformation.

How to develop your first Angular application? Take part in this hands-on workshop that introduces you to the framework and the ecosystem's essential concepts. The workshop highlights TypeScript and RxJS, which play an important role in the framework.

The workshop is very practice-oriented, and over half of it consists of practical exercises on a consistent topic. You work on all aspects of Angular based on specific tasks. After the workshop, your team can apply the acquired techniques in their daily work.

**Angular basics**

- › TypeScript
- › ng-cli

**Components**

- › Components and their lifecycle methods
- › Templates
- › Events and event handling
- › Unit testing of angular components

**Data management**

- › RxJS and observables
- › Services

**Related topics**

- › Angular modules
- › Client-side routing

**Target audience**

Software developers

**Duration**

2 days

**Prerequisites**

Experience in C++ is useful

Questions?  
workshops@  
methodpark.de

# QT/QML



Are you looking for a framework to develop cross-platform programs and graphical user interfaces quickly and clearly?

With Qt and QML, you develop platform-independent, high-performance user interfaces on Windows, Linux, iOS, and Android with reasonable effort. This hands-on workshop delivers everything you need to know.

Gain insights into the concepts and methods of interface programming with Qt & QML and receive an overview of the QML basics, UI interactions, and Qt quick controls. This workshop explains the interaction of QML & C++ code and other important topics from the Qt library.

Deepen the content with practical exercises, which take up about half of the workshop and help you understand and consolidate the acquired knowledge.

**QML basics**

- › Introduction to the Qt Creator development environment
- › Which basic elements are available?
- › What are properties, and how do property bindings work?
- › How do QML and Javascript work together?
- › How to position elements?
- › How to react to keyboard input and mouse clicks?

**The component library: Qt quick controls**

- › Introduction and use of quick controls
- › Your design adaptation of standard components
- › Differences between quick controls version 1 and 2

**Extend the QML functionality**

- › Create components and component libraries
- › Integration of C++ objects in QML
- › Create C++ QML components
- › On request: Introduction to Qt for Python

**Other related concepts:**

- › Model/view concept and data display
- › Internationalization
- › How to dynamically create elements at runtime?

**Target audience**  
Software developers

**Duration**  
2 days

**Prerequisites**  
Experience in  
software development

Questions?  
workshops@  
methodpark.de

# CLEAN CODE



Developers primarily bear responsibility for the quality of source code. Clean code offers them a set of methods to secure and improve software quality consistently.

Software gradually loses quality over time, making implementations and changes more difficult. The code becomes increasingly incomprehensible. Clean code prevents this and prepares your software for future challenges. This hands-on workshop offers the techniques and practices necessary to write coherent, extensible, and resilient code.

The workshop places great emphasis on hands-on knowledge transfer: you develop clean code based on specific exercises, and after the workshop, you can use the acquired techniques in your daily project work.



## Clean code vs. bad code

- › What is bad code?
- › What is clean code?
- › The Boy Scout Rule

## Aesthetic code

- › Why is naming so important?
- › What are clean methods?
- › What are the characteristics of a good class?
- › How to write comments correctly?

## Formats and structures

- › How to format cleanly?
- › How to structure objects and data?
- › How to use refactoring?
- › How to deal with the limits of other systems?

## Quality assurance

- › What to do with errors?
- › How to proceed with unit tests?

## SOLID

- › You basically did everything right?

## Clean code as a mentality

- › Clean code developer grades: How to internalize CCD principles and practices?

**Target audience**Software developers,  
testers**Duration**

2 days

**Prerequisites**Experience in  
software development**Questions?**workshops@  
methodpark.de

# UNIT TESTS



High-quality software needs automated testing, and unit tests lay the foundation for successful development.

Would you like to intensify the introduction of unit tests into your software development, provide a safety net for your existing application, and look for training tailored to your needs? This unit test hands-on workshop offers the required techniques and best practices.

The workshop is very practice-oriented, and half of it consists of practical exercises on a consistent topic. You acquire all aspects of unit testing based on concrete exercises.

**Basics**

- › What are unit tests?
- › Which tools are used?

**Test-first approach**

- › Test Driven Development (TDD)
- › Behavior Driven Development (BDD)

**Dealing with dependencies**

- › Test doubles/mocks
- › Dependency injection

**Unit tests in legacy environments**

- › Legacy code
- › Code coverage
- › Clean code
- › Refactoring
- › Golden master

**Target audience**

Software developers,  
build managers, integrators

**Duration**

2 days

**Prerequisites**

Experience in software  
development

Questions?

workshops@  
methodpark.de

# CONTINUOUS INTEGRATION & DEPLOYMENT



Continuous integration, delivery, and deployment are elementary components of modern software development with short release cycles and high quality.

Nightly builds are not always adequate since build and delivery pipelines quickly become extensive, and it can become difficult to define the correct build size.

This hands-on workshop offers you the techniques and practices to successfully deploy and deliver continuous integration and delivery in your project. It emphasizes practice-oriented knowledge transfer: with concrete exercises, you independently work on all aspects of continuous integration and delivery.

For the exercises, we use Jenkins or the CI system of your choice to make your organization's deployment as smooth as possible, and you can discuss other tools used in your organization in detail. After the workshop, your team can use the acquired techniques in their daily work.

**CI and CD**

- › What is continuous integration?
- › What is continuous delivery/deployment?
- › How do both work together?

**Builds**

- › How to implement a build pipeline step by step?
- › How to deal with failed builds?

**Tools**

- › Which tools support your work?
- › According to which criteria do you select your tools?

**Test, analyze & review**

- › How to integrate automatic tests?
- › How to include code analyses?
- › How to integrate code reviews?

**Target audience**

Software architects,  
software developers

**Duration**

2 days

**Prerequisites**

Basic programming  
skills

Questions?

workshops@  
methodpark.de

# BLOCKCHAIN



Blockchains, cryptography, smart contracts, DApp, decentralized networks – relevant tools & frameworks.

Have you already read and heard a lot about blockchains? But you still look for a better overview of the bulk of terms and technologies? Or are you still wondering for what to use a blockchain? Just try it out and gain practical experience in creating your first decentralized application.

With a consistent example, you learn how to use all the relevant tools and frameworks you need to create your application. You also receive numerous practical tips and tricks, which save you much time when implementing your ideas.

**Basics**

- › Blockchains and cryptography
- › What is a smart contract?
- › Differences and overview of current blockchain projects

**Create your projects**

- › The Ethereum project in detail
- › How to create smart contracts in Solidity with the IDE remix
- › Using the Truffle framework
- › Creating a frontend for smart contracts

**Quality assurance for decentralized applications**

- › What are the differences compared to conventional web applications?
- › Testing smart contracts and decentralized applications
- › Linting and static analysis tools for smart contracts
- › Security practices for smart contracts

**Design patterns, standards, and protocols**

- › Design patterns for smart contracts
- › ERC standards and ERC20/ERC721 tokens
- › Game theory and token economics
- › Analysis of existing token models
- › Complementary technologies such as IPFS or zero-knowledge proofs
- › Raiding, plasma, and the future of Ethereum

### Target audience

Designers, developers, concept developers, project managers, product owners

### Duration

2 days

### Prerequisites

With or without experience in usability/user experience

Questions?

workshops@  
methodpark.de

# USER EXPERIENCE



Good usability and a positive user experience (UX) determine the success of your products.

If product development does not meet your users' needs, they will be dissatisfied sooner or later. This dissatisfaction results in competitive disadvantages for your company.

Outstanding software products need the dialogue with their users and the integration of their use cases.

In this hands-on workshop, experienced trainers show you the theoretical basics of UX. You test the acquired methods in practical exercises and learn to apply them in your projects.

Tailor the workshop content to your needs.

- › Choose a topic from any phase of user-centric product development: analysis, conception, implementation, or testing.
- › Select the required training content from the list of offered topics.
- › Based on your specific expectations and needs, we define your UX training agenda together with you.

## User experience 101

- › User-Centered Design (UCD)
- › Definition of usability & user experience (UX)
- › Discussion of basic usability Do's and Don'ts
- › Overview of different heuristics and usability criteria
- › Overview of different development methods

## Design basics

- › Ideation (concept & design)
- › Design criteria
- › Design laws
- › Methodical generation of design ideas

## User requirements

- › Context and requirements analysis (goals and methods)
- › User needs and requirements
- › Potential user groups (personas)

## UX research

- › Research & interpretation
- › Heuristic evaluation (expert review)
- › Usability problems
- › Analysis of use case scenarios

## Prototyping

- › Overview of prototypes
- › Prototyping tools
- › Evaluation (prototyping & testing)

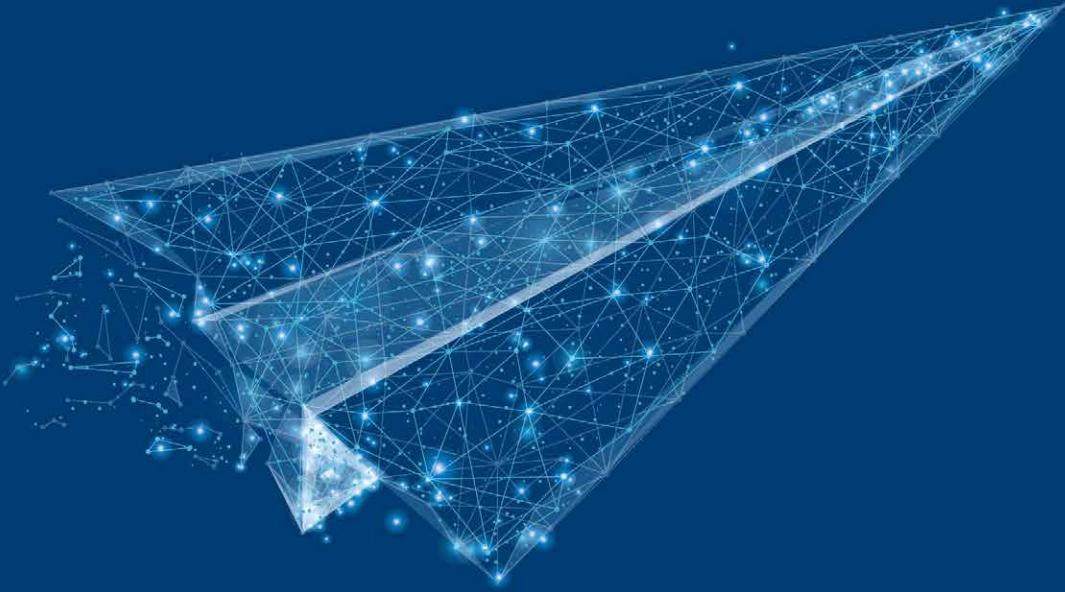
## Quality assurance

- › ISO standards relevant for usability
- › Development support

## Work with your content

- › Usability talk: joint ad-hoc reviews based on examples

# Content



**Method Park by UL**

Wetterkreuz 19a | 91058 Erlangen  
Germany

Tel. +49 9131 97206-0 | [info@methodpark.de](mailto:info@methodpark.de)

[WWW.METHODPARK.COM](http://WWW.METHODPARK.COM)