

## Training program:

# AI-Assisted SDLC - foundations and advanced techniques

### Info:

<b>Name:</b>	<b>AI-Assisted SDLC - foundations and advanced techniques</b>
<b>Code:</b>	<b>ai-assisted</b>
<b>Category:</b>	AI
	architects
	developers
<b>Target audience:</b>	team_lead
	tech_lead
	testers
<b>Duration:</b>	2 days
<b>Format:</b>	60% praktyki / 40% teorii

Master advanced AI-assisted programming techniques and transform software development. Using AI solely for code generation made sense when this technology was in its infancy. Now, in the age of agent-based AI, we'll show you how to use it as a partner throughout the entire software development process—from requirements analysis and business communication, through implementation, testing and code review, all the way to documentation.

During the training:

- You will learn how to make AI seamlessly handle the things developers hate: task descriptions, tests, logs, documentation.
- You will learn practical techniques for working with AI agents (such as Claude Code).
- You will see how to effectively manage context and integrate AI with everyday tools.
- Step by step, you will go from basic configuration to real workflows that accelerate the work of the entire team.
- You will receive an introduction to advanced topics such as sub-agents for task delegation, MCP for extending functionality, parallel processing mindset, and version control with AI.

**This is not just another course about prompts. It's a transformation in the way you work – AI becomes your extra team member, never complaining and always delivering.**

Training that will teach you how to use AI as a partner in your software development process. From basic installation to advanced context management techniques – everything you need to start working effectively with AI agents, using the example of Claude Code.

## It's all about the content.

- Real-world use cases - working on real projects
- Hands-on with Claude Code - immediate application of knowledge
- Learning by doing

# Training program

## 1. Introduction to the AI-Assisted Paradigm

- 1.1. The Evolution of AI Support in Programming
- 1.2. Mental Models and Changing Your Mindset
- 1.3. First Steps with Claude Code
- 1.4. @notation - Precise File References

## 2. CLAUDE.md Supremacy - Context as the Foundation

- 2.1. The Philosophy and Power of System Instructions
- 2.2. Workshop: Analyzing Effective CLAUDE.md
- 2.3. Workshop: Creating Your Own CLAUDE.md
- 2.4. Context Poisoning and How to Avoid It

## 3. Plan Mode, Context, and Work Documentation

- 3.1. Plan Mode - Think Before You Act
- 3.2. Context Window Management
- 3.3. STATUS.md and Context Documentation
- 3.4. Todo Lists and Progress Tracking
- 3.5. Workshop: Workflow with Full Documentation

## 4. Debugging and Tight Feedback Loops

- 4.1. AI as a Debugger
- 4.2. Tight Feedback Loops - Writing Execution Iteration
- 4.3. Workshop: Comprehensive Debugging Session
- 4.4. Context-Sized Chunking

## 5. Refactoring and Working with Legacy Code

- 5.1. A Strategic Approach to Legacy Code

5.2. Batch Operations with CLAUDE.md

5.3. Workshop: Refactoring Using CLAUDE.md

5.4. Explicit Adherence Pattern

## 6. Testing with AI and Verification

6.1. Test-Driven Development with AI

6.2. Testing Levels and Strategy

6.3. Workshop: End-to-End Testing

6.4. Bash Automation for Test Runners

## 7. Basics of Subagents and MCP

7.1. Task/Agent Tools - Delegating Specialized Tasks

7.2. MCP (Model Context Protocol) - Basics

7.3. Version Control with AI

7.4. Workshop: End-to-End Workflow with Subagents

## 8. Best Practices and Optimization

8.1. Effective Prompting Mastery

8.2. Optimizing Work Performance

8.3. Building an AI-Assisted Culture in Your Team