BO·TT·EGA
IT minds

Training program:

Harness Engineering - AI Software Factory

Info:

Name:	**Harness Engineering - AI Software Factory**
Code:	**harness-engineering**
Category:	AI
Target audience:	developers architects analysts tech_lead
Duration:	3 days
Format:	30% lecture / 70% workshop

Harness Engineering is an advanced program for organizations that want to move from occasional AI usage to a managed AI-Native SDLC operating model. The training shows how to build an AI-supported software factory: from reverse engineering of critical context, through business collaboration and rapid prototyping, to production-ready quality, troubleshooting, and CI/CD pipeline integration.

This is not a prompts or Claude Code course. It is a practical operating model for teams working in complex environments with architectural and compliance constraints, while still needing to reduce lead time and lower the cost of defects.

It's all about the content.

- How to reconstruct and systematize architectural and business context so AI can operate on reliable inputs
- How to run AI Assisted Forward Deployed Engineering with business experts, from raw material to implementation-ready specification
- How to design long-running autonomous AI workflows and split responsibilities between humans and agents
- How to implement quality gates: validation against architecture, business specification, and test quality
- How to establish a repeatable AI troubleshooting
- How to integrate AI into CI/CD: triage, implementation automation, and AI pull request review

BOTTEGA IT minds

Training program

1. Reverse Engineering of Critical Context

- 1.1. Defining architectural context and the patterns used in the project
- 1.2. Documenting business context
- 1.3. Building an index of key elements in external systems and in the system under development
- 1.4. Creating a minimal context pack for AI agents: what must be explicit, up-to-date, and versioned

2. AI Assisted Forward Deployed Engineer

- 2.1. Three-way collaboration model: business, engineering, AI
- 2.2. Processing transcripts, notes, and raw business materials
- 2.3. Gap analysis: identifying gaps between business intent and implementation readiness
- 2.4. Rapid prototyping and fast stakeholder validation
- 2.5. Defining and evaluating specification quality (Definition of Ready)
- 2.6. Change impact analysis: forecasting impact on scope, integrations, and tests

3. Development Workflow

- 3.1. Designing long-running autonomous AI workflows for real engineering tasks
- 3.2. Demonstrating the effectiveness of architectural instructions through implementation outcomes
- 3.3. Enforcing code compliance with adopted architectural patterns and design standards
- 3.4. Optimizing the Human-AI responsibility split in day-to-day delivery

4. Quality Gates and AI Review

- 4.1. Bug-reduction strategies before human code review starts
- 4.2. Validating changes against architecture and technical constraints
- 4.3. Validating changes against business specification and acceptance criteria
- 4.4. Evaluating test quality: risk coverage, critical scenarios, and test anti-patterns

BOTTEGA IT minds

5. AI Troubleshooting

5.1. How to instruct AI to work effectively with databases

5.2. How to instruct AI to work effectively with APIs

5.3. Local deployment and monitoring as a tight feedback loop

5.4. Standard troubleshooting flow

6. Integration with Pipelines

6.1. Integrating AI with CI/CD pipelines and quality policies

6.2. Automatic task triage and work prioritization

6.3. Automated implementation of selected task classes within governance boundaries

6.4. Automated AI PR review with human escalation for ambiguous cases