Course Specification: Certified Scrum Developer

Learning Objectives

- **Lean, Agile, and Scrum Fundamentals**
- **Definition of Done**, a key practice in ensuring quality standards are met.
- **Effective Team Dynamics**, the difference between true teams and working groups, and how to develop as a team.
- **Emergent Architecture and Design**, approaches that allow designs and architecture to emerge and change over time.
- **Test-First Development**, building quality into the development process rather than relying only on checking for errors later.
- **Refactoring** as a practice to continually increase quality and built-in maintainability.
- **Continuous Integration** and code ownership to detect integration errors as quickly as possible.

And should be certified as a **Scrum Developer**.

Couse outline:

1. Lean, Agile, and Scrum Fundamentals

- Learn the underpinning principles of Lean, Agile, and Scrum to better understand how technical practices help in product delivery.
- Learning how technical practices support and enhance product delivery within

2. Definition of Done

- Learning facilitation techniques to help teams create and apply a solid DoD.
- Coaching teams to consistently adhere to shared quality standards.

3. Effective Team Dynamics

- Understanding what differentiates a true team from a working group.
- Developing skills to support the team's growth stages and collaboration patterns.
- Exploring the Scrum Master's role as a Servant-Leader to promote a healthy team environment.

4. Test-First Development

- Applying Test-First techniques to build quality into the development process early.
- Understanding how upfront testing reduces defects and enhances maintainability.
- Improving developer discipline by writing tests before code to guide design.

5. Refactoring

- Practicing continuous refactoring to improve code structure without altering behaviour.
- Enhancing long-term maintainability through incremental improvements.
- Reducing technical debt by routinely revisiting and optimizing existing code.

6. Emergent Architecture and Design

- Applying development approaches that allow architecture and design to evolve over time.
- Leveraging iterative feedback to refine architectural decisions as the product grows.
- Supporting adaptive design to align with changing requirements.

7. Continuous Integration

- Using Continuous Integration (CI) to detect integration issues early.
- Encouraging shared code ownership to support a stable and frequently integrated.

Duration: 2-4 days
Exam: Yes
Certification: Certified Scrum Developer
Attendance: Face-To-Face or On-line (face-to-face is preferred)