Communist Manifesto for Agile Project Deliverables Development

A Marxist-Leninist adaptation of the Manifesto for Agile Project Deliverables Development, seen here: Agile Manifesto

We are uncovering better ways of developing project deliverables by doing it and helping others do it. Through this work we have come to value:

- **Symbiotic relationships between individuals and collectives** over **unilateral orders**

- **Working project deliverables** over **comprehensive documentation**

- **Working class collaboration** over **individualized piecework**

- **Responding to change** over **following a preset plan**

All of these items are valuable to their separate social systems, but the items on the **left** will allow responsive development that is relevant to the current material conditions of working-class society.

12 Principles of AgCom Project Deliverables

1. Our highest priority is to empower the working class through early and continuous project deliverables prioritized by value added and/or magnitude of impact.

2. Welcome changing requirements, even late in development. Dialectical materialism, like Agile processes, is about anticipating and responding to change, in order to harness the power of the working class.

3. Deliver working project deliverables frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Developers, working people, unions, and representatives must work together daily throughout the project.

5. Build projects that motivate people, and create an environment and the support to let them get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working project deliverables are the primary measures of progress.

8. AgCom processes promote sustainable development. Developers, working people, unions, and representatives should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity, the art of maximizing the amount of work not done, is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.Pight to