

Traditional done well

Work as a team

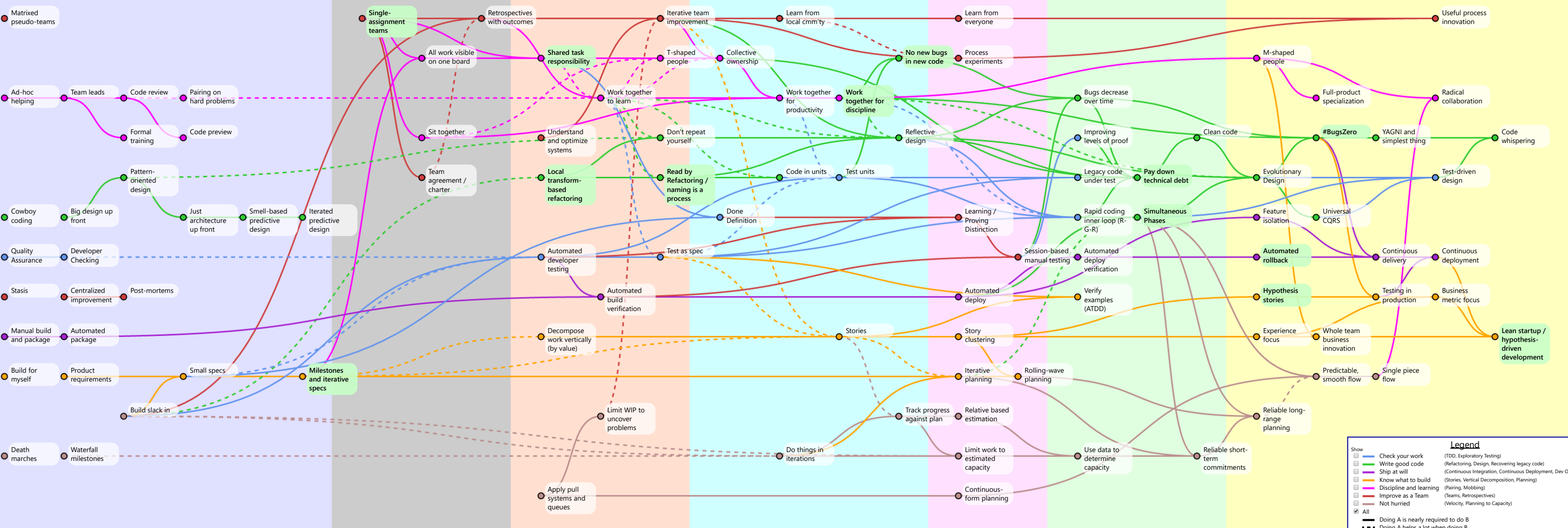
Core modern engineering

Adaptive engineering

Leverage capabilities

Eliminate debt and change the rules

Take advantage of being awesome



Show

- Check your work (TDD, Exploratory Testing)
- Write good code (Refactoring, Design, Recovering legacy code)
- Ship at will (Continuous Integration, Continuous Deployment, Dev Ops)
- Know what to build (Stories, Vertical Decomposition, Planning)
- Discipline and learning (Pairing, Mobbing)
- Improve as a Team (Teams, Retrospectives)
- Not hurried (Velocity, Planning to Capacity)
- All

Legend

- Doing A is nearly required to do B
- - - Doing A helps a lot when doing B