How to Structure a Low-Code Team
Take a "Developer And" Approach.

Low-code is the new way for IT teams to build applications. It helps streamline development with nimble, balanced teams that minimize dependencies and speed time to value.

The best low-code teams are full of "developer and _____" roles. Think "developer and tester" or "developer and Scrum Master." Everyone from team leads to architects can develop features. And every developer should have at least one additional specialty area, such as UX design, testing, or business analysis.

As you structure your low-code team, include these six specialty areas:

Product Owner

The product owner should be the only role that does not also develop features. They guide the team to build the right things, expressing what's needed from the business, prioritizing value, and ensuring requirements are visible and clear.
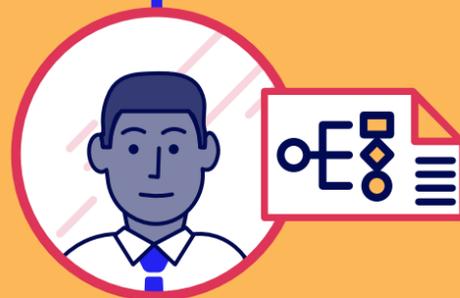
Developer and **UX Designer**

The person covering UX design ensures good design principles are followed and crafts solution blueprints for important design decisions.

Developer and **Architect**

This role makes sure the team builds the application the right way by directing the solution architecture.

Developer and **Business Analyst**

This role will dig into the business requirements and user needs. They ask probing questions and think through every scenario to turn complex processes into something that can be coded in a solution.

Developer and **Tester**

At least one team member should focus on exploratory testing to find breakable areas that are missed by test scripts. They also have knowledge of tools for automated tests and performance and load testing.

Developer and **Team Lead or Scrum Master**

This role enables the team to build quickly. They'll coordinate development work, removing impediments and encouraging agile practices.

Rethink your approach to low-code.

To see the full power of low-code, trade outdated development approaches for some key agile practices. Read about common mistakes and how to rethink your approach in this eBook: 4 Pitfalls of Low-Code Application Development.