

Domain-Driven Design (DDD)

- For ...
 - complex needs
 - connects implementation to evolving model
 - **DOMAIN** is the problem space
 - **MODEL** is an abstracted potential solution
 - Code is code

DDD: From Debt to Dividend

- **Tech Debt**
 - Paying interest plus principle to temporarily move faster
- **Tech Dividend**
 - Getting extra value on deep design to unlock extra features