

AGILE ARCHITECTURE AND TECHNICAL DESIGN

Presentation Summary



Agile Architecture

- Waterfall has specific phases for architecture and technical design
- Scrum doesn't call out these phases
- **Big design up front (BDUF)** isn't compatible with agile due to evolving requirements
- The **last responsible moment** defers technical decisions as late as possible
- Avoids **YAGNI**: *You Ain't Gonna Need It*
- Scrum compresses the waterfall phases—you do **a little bit of everything all the time**
- Some up-front decisions are required: technology, frameworks, languages, patterns, practices

Walking Skeleton

- Lightweight application framework
- Crystallizes up-front decisions, but doesn't include product-specific functionality
- Architecture and functionality evolve in parallel

Scrum Development Team

- Development team owns technical decisions
- Architecture becomes **emergent**
- It's OK to think long-term, but only **act short-term**

Supporting the Scrum Team

- **Inversion of power** ensures the organization is acting in service to the Scrum team
- Leverage internal knowledge to help the team
- Some companies create **patterns and practices** teams to spread best practices
- Instead of, "Do it this way," leaders should try, "You might want to consider" or, "You might want to think about"
- The key is to **help** the Scrum team without **taking ownership** away from them