SWE 4743:
Object-Oriented Design

Jeff Adkisson

Single Responsibility Principle (SRP)

The 'S' in SOLID

Cohesion, Coupling, and Responsibility

Lecture Material

See **04-single-responsibility-principle.md** in the Presentations folder for the detailed lecture material.

This slide deck is a bare-bones AI-generated summary of the detailed source material.

Introducing SOLID

- Foundational object-oriented design principles
- Popularized by Robert C. Martin
- Focus on managing long-term change

Designing for Change

- Architecture exists to support change
- Code can run without architecture
- Good architecture reduces change friction

SRP: The Cornerstone of SOLID

- SRP underpins all other SOLID principles
- Defines responsibility boundaries
- Focuses on why code changes

The Original Definition of SRP

- One reason to change
- Simple but ambiguous
- Led to inconsistent interpretations

Redefining Responsibility with Actors

- Actors represent sources of change
- Conceptual, not technical
- One module serves one actor

The Real Risk of SRP Violations

- Hidden coupling
- Fragile code
- Unintended side effects

Change Amplification

- Small changes cause widespread impact
- Multiple teams affected
- Indicates broken boundaries

SRP vs DRY

- DRY reduces duplication
- SRP isolates reasons to change
- Apply SRP before DRY

Cohesion and Coupling

- High cohesion groups related behavior
- Low coupling limits dependencies
- SRP drives balance

SRP Across Levels of Code Organization

- Applies at all levels
- Classes to systems
- Fractal principle

Screaming Architecture

- Architecture communicates intent
- Business features over frameworks
- Improves comprehension

Feature-First Layered Architecture

- Organize by feature first
- Layer internally where useful
- Supports evolution

Why SRP Is Hard to Apply

- Not visible in syntax
- Problems appear over time
- Requires judgment

Conclusion

- SRP manages change
- Aligns code with organization
- Enables sustainable systems