

The Acyclic Dependency Principle

Unentangle development and integration

- Problem: ‘the morning after syndrome’
 - When many developers are modifying the same source files
- Solution One: Weekly Build
 - Develop independently, integrate regularly
 - Integration burden can grow with the project size, leading to high risks
- Solution Two: Releasable Components
 - Partition the development into releasable components
 - Dependencies upon components don’t cause a cycle

The Acyclic Dependency Principle

No cycle in the component dependency graph

- A cycle in the component dependency graph
 - No way to decide the order of building components
 - Very hard to isolate components, unit-test and release
- Break the cycle
 - Apply the Dependency Inversion Principle (DIP)
 - Create a new abstract component
- Component structure cannot be designed from the top down
 - Different from functional decomposition (top-down design)